

PHME 2022

Proceedings of the
7th European Conference of the
Prognostics and Health Management Society
2022

Turin, Italy
July 6th - July 8th , 2022

ISBN – 978-1-936263-36-3

Edited by:

Phuc Do

Gabriel Michau

Cordelia Ezhilarasu



Management team

General operational functions:

Steve King – Cranfield University – General Chair
Ian Jennions – IVHM Centre, Cranfield University – Vice-Chair
Octavian Niculita – Glasgow Caledonian University – Finance
Claude Fourbert – VERT COM – General Support chair

External affairs:

Jeff Bird – TECnos – Sponsorship Chair
Claude Fourbert – VERT COM – Website Chair

Specific sessions:

Ryan Walker – Mercedes F1 – Panel Chair
Bin Zhang - University of South Carolina – Panel Co-Chair
Danilo Giordano – Politecnico di Torino – Data Challenge Chair
Martini Trevisan – Politecnico di Torino – Data Challenge Co-Chair
Kamal Medjahar – Tarbes National School of Engineering – Doctorial Symposium Chair
Khanh Nguyen – Tarbes National School of Engineering, Doctorial Symposium Co-Chair
Ferhat Tamssaouet - University of Perpignan, Doctorial Symposium Co-Chair
Phuc Do – University of Lorraine - Doctorial Symposium Co-Chair
Tingting Zhu - Oxford University – Tutorial Chair

Technical content:

Gabriel Michau – Stadler Service AG – TPC Chair
Cordelia Ezhilarasu - IVHM Centre, Cranfield University – TPC Co-Chair
Phuc Do – University of Lorraine – Proceedings Chair

Technical Program Committee members:

Zeina Al Masry – Femto, France	Yanfu Li - Tsinghua University, China
Manuel Arias Chao - ZHAW, China	Gabriel Michau - Stadler Service AG, Switzerland
Piero Baraldi – Polimi, Italy	Ahmed Mosallam – Schlumberger, France
Christophe Berenguer - Grenoble University, France	Octavian Niculita - Glasgow Caledonian University, UK
Oliver Cassebaum - Volkswagen, Germany	Khanh Nguyen - Tarbes National School of Engineering, France
Pierre Dersin - Lulea University, Sweden	Slawomir Nowaczyk - Halmstad University, Sweden
Phuc Do - University of Lorraine, France	Marcos Orchard - University of Chile, Chile
Cordelia Ezhilarasu - Cranfield University, UK	Sepideh Pashami - Halmstad University, Sweden
Fink Olga - EPFL, Switzerland	Bruce Stephen - Strathclyde University, UK
Kareem Gouda - SKF, Netherlands	Alexandre Voisin - University of Lorraine, France
Henss Mark - Stuttgart University, Germany	Dong Wang - Shanghai Jiao Tong University, China
Benoit Iung - University of Lorraine, France	Ferhat Tamssaouet - University of Perpignan, France

Published by **PHM Society**

Publisher Address:

241 Woodland Drive, State College, PA 16803

<http://www.phmsociety.org/journal/publisher>

Table of contents

- 1 Experiences of a Digital Twin Based Predictive Maintenance Solution for Belt Conveyor Systems
Kammal Al-Kahwati, Wolfgang Birk, Evert Flygel Nilsfors, Rune Nilsen
- 9 A Case-study Led Investigation of Explainable AI (XAI) to Support Deployment of Prognostics in the Nuclear Industry.
Omnia Amin, Blair Brown, Bruce Stephen, Stephen McArthur
- 21 Long Horizon Anomaly Prediction in Multivariate Time Series with Causal Autoencoders
Mulugeta Weldezigina Asres, Grace Cummings, Aleko Khukhunaishvili, Pavel Parygin, Seth I. Cooper, David Yu, Jay Dittmann, Christian W. Omlin
- 32 Experimental Validation of Multi-fidelity Models for Prognostics of Electromechanical Actuators
Leonardo Baldo, Pier Carlo Berri, Matteo D. L. Dalla Vedova, Paolo Maggiore
- 43 An Analysis of Vibrations and Currents for Broken Rotor Bar Detection in Three-phase Induction Motors
Amirhossein Berenji, Zahra Taghiyarrenani
- 49 Online Flow Estimation for Condition Monitoring of Pumps in Aircraft Hydraulics
Phillip Bischof, Frank Thielecke, Dirk Metzler
- 58 Hybrid Fault Prognostics for Nuclear Applications: Addressing Rotating Plant Model Uncertainty
Jennifer Blair, Bruce Stephen, Blair Brown, Alistair Forbes, Stephen McArthur
- 68 Data-driven Prognostics based on Evolving Fuzzy Degradation Models for Power Semiconductor Devices
Khoury Boutrous, Iury Bessa, Vicenç Puig, Fatiha Nejjari, Reinaldo M. Palhares
- 78 State of Health and Lifetime Prediction of Lithium-ion Batteries Using Self-learning Incremental Models
Murilo Camargos, Plamen Angelov
- 87 Wrong Injection Detection in a Small Diesel Engine, a Machine Learning Approach
Piero Danti, Giovanni Vichi, Ryota Minamino
- 96 Novel Metrics to Evaluate Probabilistic Remaining Useful Life Prognostics with Applications to Turbofan Engines
Ingeborg de Pater, Mihaela Mitici
- 110 Filtering Misleading Repair Log Labels to Improve Predictive Maintenance Models
Pablo del Moral, Sławomir Nowaczyk, Sepideh Pashami
- 118 Physics-informed lightweight Temporal Convolution Networks for fault prognostics associated to bearing stiffness degradation
Weikun Deng, Khanh T. P. Nguyen, Christian Gogu, Jérôme Morio, Kamal Medjaher
- 126 Design and validation of scalable PHM solutions for aerospace onboard systems
Fabio Federici, Cecilia Tonelli, Mathieu Le Cam, Marcello Torchio, David Larsen
- 136 Sensor fault/failure correction and missing sensor replacement for enhanced real-time gas turbine diagnostics
Amare Fentaye, Valentina Zaccaria, Konstantinos Kyprianidis
- 146 Helicopter Bolt Loosening Monitoring using Vibrations and Machine Learning
Eli Gildish, Michael Grebshtein, Yehudit Aperstein, Alex Kushnirski, Igor Makienko
- 156 On the Integration of Fundamental Knowledge about Degradation Processes into Data-Driven Diagnostics and Prognostics Using Theory-Guided Data Science
Simon Hagmeyer, Peter Zeiler, Marco F. Huber
- 166 Toward Runtime Assurance of Complex Systems with AI Components
Yuning He, Johann Schumann, Huafeng Yu
- 175 Machine Learning Methods for Health-Index Prediction in Coating Chambers
Clemens Heistracher, Anahid Jalali, Jurgen Schneeweiss, Klaudia Kovacs, Catherine Laflamme, Bernhard Haslhofer
- 182 Approximate Bayesian Computation as a New Tool for Partial Discharge Analysis of Partial Discharge Data
Kai Hencken, Daniele Ceccarelli, Elsi-Mari Borrelli, Andrej Krivda
- 193 Unsupervised Prognostics based on Deep Virtual Health Index Prediction
Martin Hervé de Beaulieu, Mayank Shekhar Jha, Hugues Garnier, Farid Cerbah

- 200 Autoencoder based Anomaly Detection and Explained Fault Localization in Industrial Cooling Systems
Stephanie Holly, Robin Heel, Denis Katic, Leopold Schoeffl, Andreas Stiftinger, Peter Holzner, Thomas Kaufmann, Bernhard Haslhofer, Daniel Schall, Clemens Heitzinger
- 211 Joint Autoencoder-Classifer Model for Malfunction Identification and Classification on Marine Diesel Engine Diagnostics Data
Kurçat Ince, Gazi Koçak, Yakup Genc
- 219 Physics Informed Neural Network for Health Monitoring of an Air Preheater
Vishal Jadhav, Anirudh Deodhar, Ashit Gupta, Venkataramana Runkana
- 231 A Health Index Framework for Condition Monitoring and Health Prediction
Alexander Athanasios Kamtsiuris, Florian Raddatz, Gerko Wende
- 239 Tool Compatibility Index: Indicator Enables Improved Tool Selection for Well Construction
Jinlong Kang, Christophe Varnier, Ahmed Mosallam, Noureddine Zerhouni, Fares Ben Youssef, Nannan Shen
- 245 An End-to-End Pipeline for Uncertainty Quantification and Remaining Useful Life Estimation: An Application on Aircraft Engines
Marios Kefalas, Bas van Stein, Mitra Baratchi, Asteris Apostolidis, Thomas Back
- 261 Fault Detection in a Wind Turbine Hydraulic Pitch System Using Deep Autoencoder Extracted Features
Panagiotis Korkos, Jaakko Kleemola, Matti Linjama, Arto Lehtovaara
- 269 iVRIDA: intelligent Vehicle Running Instability Detection Algorithm for high-speed rail vehicles using Temporal Convolution Network – A pilot study
Rohan R. Kulkarni, Rocco Libero Giossi, Prapanpong Damsongsaeng, Alireza Qazizadeh, Mats Berg
- 278 Remaining-Useful-Life prognostics for opportunistic grouping of maintenance of landing gear brakes for a fleet of aircraft
Juseong Lee, Ingeborg de Pater, Stan Boekweit, Mihaela Mitici
- 286 Novel Graph-Based Features for Bearing Fault Diagnosis: Two Aspects of Time Series Structure
Sangho Lee, Chihyeon Choi, Youngdoo Son
- 294 Certainty Groups: A practical approach to distinguish confidence levels in neural networks
Lukas Lodes, Alexander Schiendorfer
- 306 Processing of Condition Monitoring Annotations with BERT and Technical Language Substitution: A Case Study
Karl Lowenmark, Cees Taal, Joakim Nivre, Marcus Liwicki, Fredrik Sandin
- 315 A Design Methodology for Robust Model-Based Fault Diagnosis Schemes and its Application to an Aircraft Hydraulic Power Package
Felix Mardt, Phillip Bischof, Frank Thielecke
- 329 Prognosis of wear progression in electrical brakes for aeronautical applications
Andrea De Martin, Giovanni Jacazio, Vincenzo Parisi, Massimo Sorli
- 338 Domain knowledge informed unsupervised fault detection for rolling element bearings
Douw Marx, Konstantinos Gryllias
- 351 Estimation of Wind Turbine Performance Degradation with Deep Neural Networks
Manuel S. Mathew, Surya Teja Kandukuri, Christian W. Omlin
- 360 Weighted-QMIX-based optimization for maintenance decision-making of multi-component systems
Van-Thai Nguyen, Phuc Do, Alexandre Voisin, Benoit Iung
- 368 Data Driven Seal Wear Classifications using Acoustic Emissions and Artificial Neural Networks
Nadia S. Noori, Vignesh V. Shanbhag, Surya T. Kandukuri, Rune Schlanbusch
- 376 Severity Estimation of Faulty Bearings Based on Strain Signals From Physical Models and FBG Measurements
Ravit Ohana, Renata Klein, Jacob Bortman
- 384 A Comparative Study of Health Monitoring Sensors based on Prognostic Performance
Hyung Jun Park, Nam Ho Kim, Joo-Ho Choi
- 392 Forecasting piston rod seal failure based on acoustic emission features in ARIMA model
Jørgen F. Pedersen, Rune Schlanbusch, Vignesh V. Shanbhag

- 401 Improved time-frequency representation for non-stationary vibrations of slow rotating machinery
Cédric Peeters, Andreas Jakobsson, Jérôme Antoni, Jan Helsen
- 410 Towards data reliability based on triple redundancy and online outlier detection
Sylvain Poupry, Cédric Béler, Kamal Medjaher
- 421 Expert Knowledge Induced Logic Tensor Networks: A Bearing Fault Diagnosis Case Study
Maximilian-Peter Radtke, Jürgen Bock
- 432 Domain adaptation in predicting turbocharger failures using vehicle's sensor measurements
Mahmoud Rahat, Peyman Sheikholharam Mashhadi, Sławomir Nowaczyk, Thorsteinn Rognvaldsson, Atabak Taheri, Ataollah Abbasi
- 440 Experimental assessment of a broadband vibration and acoustic emission sensor for rotorcraft transmission monitoring
Cristobal Ruiz-Carcel, Andrew Starr, Arturo Francese
- 449 Optical Cutting ToolWear Monitoring by 3D Geometry Reconstruction
Rob Salaets, Valentin Sturm, Ted Ooijevaar, Veronika Putz, Julia Mayer, Abdellatif Bey-Temsamani
- 458 Data-Driven Fault Detection for Transmitter in Logging-While-Drilling Tool
Karolina Sobczak-Oramus, Ahmed Mosallam, Caner Basci, Jinlong Kang
- 466 Autonomous Bearing Tone Tracking Algorithm
Alon Sol, Eyal Madar, Jacob Bortman, Renata Klein
- 473 Noise-robust representation for fault identification with limited data via data augmentation
Zahra Taghiyarrenani, Amirhossein Berenji
- 480 Automating Critical Surface Identification and Damage Detection Using Deep Learning and Perspective Projection Methods
Gautam Kumar Vadisala, Anurag Singh Rawat, Abhishek Dubey, Gareth Yen Ket Chin, Fabio Abreu
- 490 State of Health Forecasting of Heterogeneous Lithium-ion Battery Types and Operation Enabled by Transfer Learning
Friedrich Von Bulow, Tobias Meisen
- 509 Failures Mapping for Aircraft Electrical Actuation System Health Management
Chengwei Wang, Ip-Shing Fan, Stephen King
- 521 An Approach to Condition Monitoring of BLDC Motors with Experimentally Validated Simulation Data
Max Weigert
- 530 Uncertainty Informed Anomaly Scores with Deep Learning: Robust Fault Detection with Limited Data
Jannik Zraggen, Gianmarco Pizza, Lilach Goren Huber

Data challenge

- 541 Hierarchical XGBoost Early Detection Method for Quality and Productivity Improvement of Electronics Manufacturing Systems
Alexandre Gaffet, Nathalie Barbosa Roa, Pauline Ribot, Elodie Chanthery, Christophe Merle
- 550 Application of Machine Learning Methods to Predict the Quality of Electric Circuit Boards of a Production Line
Immo Schmidt, Lorenz Dingeldein, David Hunemohr, Henrik Simon, Max Weigert
- 556 A Novel Methodology for Health Assessment in Printed Circuit Boards
John Taco, Prayag Gore, Takanobu Minami, Pradeep Kundu, Alexander Suer, Jay Lee
- 563 Prediction of Production Line Status for Printed Circuit Boards
Haichuan Tang, Yin Tian, Junyan Dai, Yuan Wang, Jianli Cong, Qi Liu, Xuejun Zhao, Yunxiao Fu

Doctoral Symposium

- 571 Deep learning representation pre-training for industry 4.0
Alaaeddine Chaoub, Christophe Cerisara, Alexandre Voisin, Benoit Iung
- 574 Physics Informed Self Supervised Learning For Fault Diagnostics and Prognostics in the Context of Sparse and Noisy Data
Weikun Deng, Khanh T. P. Nguyen, Kamal Medjaher

- 577 A Novel Way to Apply Transfer Learning to Aircraft System Fault Diagnosis
Lilin Jia, Cordelia Mattuvarkuzhali Ezhilarasu, Ian Jennions
- 580 The Application, Utility and Acceptability of Data Analytics in Safety Risk Management of Airline Operations
Washington Mhangami, Stephen King, David Barry
- 583 Diagnosis and fault-tolerant control for a multi-engine cluster of a reusable launcher with sensor and actuator faults
Renato Murata, Louis Thioulouse, Julien Marzat, Hélène Piet-Lahanier, Marco Galeotta, François Farago
- 586 Artificial-intelligence-based maintenance scheduling for complex systems with multiple dependencies
Van-Thai Nguyen, Phuc Do, Alexandre Voisin, Benoit Lung
- 590 Contribution to the design and implementation of a reflexive cyber-physical system: application to air quality prediction in the valles des gaves
Sylvain Poupry, Cédric Béler, Kamal Medjaher
- 594 Combining Knowledge and Deep Learning for Prognostics and Health Management
Maximilian-Peter Radtke, Jürgen Bock
- 599 **Index of Authors**

Experiences of a Digital Twin Based Predictive Maintenance Solution for Belt Conveyor Systems

Kammal Al-Kahwati¹, Wolfgang Birk², Evert Flygel Nilsfors³, and Rune Nilsen⁴

^{1,2} *Predge AB, Vastra Varvsgatan 11, 97236 Lulea, Sweden*
Kammal.Al-kahwati@predge.se, wolfgang.birk@predge.se

² *Lulea University of Technology, Automatic Control, 97187 Lulea, Sweden*
wolfgang.birk@ltu.se

^{3,4} *LKAB Norge AS, Bolagsgata 40, 8514 Narvik Norway*
evert.nilsfors@lkab.com, rune.nilsen@lkab.com

ABSTRACT

Availability of belt conveyor systems is essential in production and logistic lines to safeguard production and delivery targets to customers. In this paper, experiences from commissioning, validation, and operation of an interactive predictive maintenance solution are reported. The solution and its development is formerly presented in Al-Kahwati et al. (Al-Kahwati, Saari, Birk, & Atta, 2021), where the principles to derive a digital twin of a typical belt conveyor system comprising component-level degradation models, estimation schemes for the remaining useful life and the degradation rate, and vision-based hazardous object detection.

Furthermore, the validation approach of modifying the belt conveyor and thus exploiting the idler misalignment load (IML) for the degradation predictions for individual components (including long-lasting ones) together with the actionable insights for the decision support is presented and assessed. Moreover, the approach to testing and validation of the object detection and its performance is assessed and presented in the same manner. An overall system assessment is then given and concludes the paper together with lessons learned.

As pilot site for the study a belt conveyor system at LKAB Narvik in northern Norway is used.

1. INTRODUCTION

Conveyor belt systems are of utmost importance in many production lines and are critical in securing the material flow between processing units. Disruption of operation due to unplanned stops when critical component failures occur can lead to unplanned costs and in worst case lead to disrupting the production and delivery to end customer. The systems are

Kammal Al-Kahwati et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

used in virtually all process industries, but also sectors where bulk material must be transported. In mining operations, they are found throughout the whole production chain.

These systems are exposed to several hazards that affect their operation and component life. Especially in the mining and haulage sector, the components degrade relatively fast compared to indoor process industries due to harsh environments such as dust, humidity, and excessive loading. In cold environments, bulk material often freezes into big blocks, and fine material can clump together due to the humid environment. There is a risk of steel plates supporting silos to get loose due to numerous collisions and ripping the rubber belt.

Condition monitoring solutions have been proposed since a long time to monitor components and their degradation and strategies on addressing these issue in a maintenance context have been suggested in (Lodewijks, 2004) and (Lodewijks & Ottjes, 2005). A difficulty in the monitoring of such systems is their distribution over a large geographic space and the sheer large number of components which are usually not equipped with sensors. In (Lodewijks, Li, Pang, & Jiang, 2016), a solution is proposed using an IoT approach with intelligent rollers. But the exchange rate of rollers in a belt conveyor system is high which means smart or intelligent versions of rollers need to be replaced and registered appropriately in an asset management registry, posing a challenge on the maintenance and IT organization. Assuming these challenges can be solved, component-level predictive maintenance would be enabled as discussed in (Liu, Pang, Lodewijks, & He, 2018) and (Liu, Pei, Lodewijks, & Zhao, 2019).

In process industries, the roll-out of component-level strategies is still in its infancy and the organizational challenges associated with it need to be first overcome. That being the case, Al-Kahwati et al. (Al-Kahwati et al., 2021) have proposed a hybrid approach to provide a component-level condition monitoring solution which does not rely on smart or

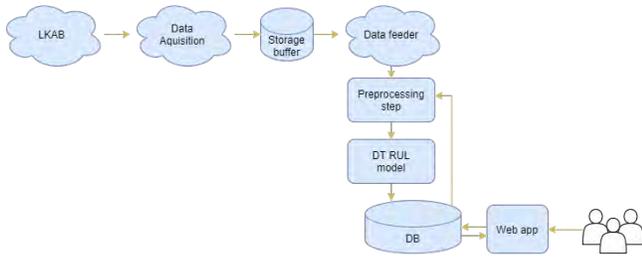


Figure 1. Simplified software architecture for the component condition monitoring model.

intelligent rollers, but is ready for such components. The solution is based on a digital twin of a belt conveyor system, extracted from CAD drawings and adjusted using field observation, comprising degradation models to quantify the component-level estimation of the remaining useful life and rate of change of the component condition. The digital twin makes use of sensor and actuator signals from the control system and static data experimental campaigns, and is realized as Software as a Service (SaaS) on a cloud-based architecture that directly integrates with the IT systems of the asset sites. Adding to this, an extension to the system is developed and commissioned, comprising of a vision based object detection system that is realised as an edge solution on the asset site.

The digital twin and the degradation modeling approach are tested and validated for a period of three months with modifications made to the belt conveyor to exploit the IML for a faster degradation rate of selected components. In addition to this, the accompanying object detection system is evaluated and validated during the same period of time. Adding to this, an approach of placing objects on the belt itself during life operation is employed for a faster evaluation and re-tuning of the detection scheme. Using the experiences collected from the field, it will be discussed how commissioning and validation of the digital twin are affected by inaccuracies in drawings in relation to the built conveyor (as-designed in contrast to as-built) and how the accuracy in reporting of maintenance actions on the decision support provided by such a solution. Moreover, how the prediction quality provided by the digital twin is affected. Some mitigation approaches for these negative effects are suggested. Furthermore, the validation approach of modifying the belt conveyor and thus exploiting IML for the degradation predictions for individual components (including long-lasting ones) together with the actionable insights for the decision support is presented and assessed based on the experiments conducted on a pilot site at LKAB Narvik in northern Norway.

2. SYSTEM SETUP

2.1. UI/UX

The component degradation model does not require new sensor installations other than those already present in the plant.

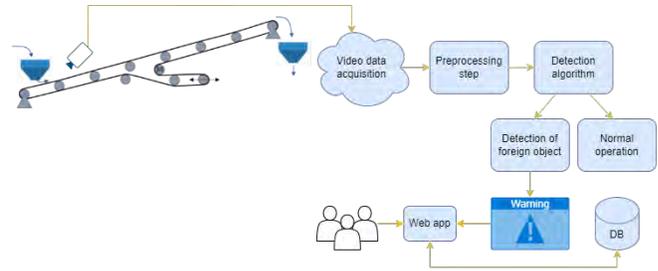


Figure 2. Software architecture for the vision-based object detection. All steps before and including the warning are done on the asset site, via the edge solution.

It is integrated with the IT infrastructure of the site and makes use of sensors readings used in the SCADA, DCS and maintenance systems. The object detection system however needs at least a camera installation if there is none present. An edge computer solution is not required but was chosen to reduce the need and risks of streaming data outside of the plant.

2.2. Component condition monitoring

The digital twin needs to be properly calibrated to the real life belt conveyor system. Information needed for capturing the dynamics and calibrating the model can be found in documents such as the function description, product information documents and CAD drawings provided by the belt conveyor system provider. In addition, the component information and data sheets from various component suppliers are needed.

Figure 1 shows a simplified version of the software architecture for the reader. Data tags from the asset site are needed to form the model for calculating the degradation of remaining useful life. The minimum requirement of data tags for the condition monitoring of the solution are the feed rate, belt speed and the belt tension as described in (Al-Kahwati et al., 2021). The data points are provided to the online cloud solution through a data lake. To have the data points available for provisioning, the tags are transferred to the data lake from the control systems.

The data lake comprises a data compression which is based on signal variation properties, like deviation from the latest registered data point. The minimum deviations needs to be properly adjusted to sufficiently register data points for replicating the system dynamics, while not reacting on noise or random behaviour. Clearly, the data can not be used for in depth data analysis on the sensor behaviour, noise properties and small signal behaviour.

Tags are posted from the data lake via a REST API to a container at Predge. The measurements are then merged into a single file which is moved for further processing and the individual measurements are backed up. When the application is ready, a data feeder will pick up the file and further process



Figure 3. Side view menu of dashboard and belt status overview. The idlers are sectioned appropriately to prevent from cluttering. Each section contains 50 idler sets, that is left, center, right roller each.

it, extracting features to determine the forces acting on the rollers through the degradation model. The remaining useful life (RUL) for each component is updated and operators can view the status of in the web application by clicking the status overview, which is shown in Figure 3.

2.3. Edge solution for object detection

Setup for the vision based object detection system is straightforward. The camera is connected and powered through an RJ45 ethernet cable and allow for using common ethernet protocols. The camera used at the asset site is a FLIR AX8 and comes with a built in light system, however depending on the current light conditions and distance to the belt, an extra light source is recommended and was used at the belt conveyor in question.

Figure 2 shows a simplified version of the software architecture of the object detection scheme. The Object detector is set up by using a computer server box set up at the asset site and captures data from a PoE camera streaming video to the local network. An AI solution for object detection is deployed as a docker container to the server box. When detections are made, the detection frames are saved and presented under the "object detection" tab in the side view menu. There the operation personnel can review all detections and mark them as false positive should the detection not be related to any object. In other words, the frames get labelled and can thus be reused for tuning or relearning purposes of detection models.

3. DECISION MAKING IN CONDITION-BASED AND PREDICTIVE MAINTENANCE

The purpose of the application and its development is to help users plan their maintenance actions based on estimations in the RUL model and detections that the object detector has made. Figure 3 shows the current status of the belt conveyor system. The carrying side of the belt is sectioned into 11 sections, containing 50 idler sets each. The colour of the section represents the worst performing roller in the section. Thresholds are set as follows:

- Green: RUL above 4000h
- Yellow: RUL between 2000-4000h
- Red: RUL below 2000h

The values are chosen to fit into current maintenance practices and to enable the user to plan ahead.

3.1. Test cases

In the proposal of the project, several KPIs were defined to assess the validity of the solution models and to provide a preliminary performance guideline for commercial use. The KPIs needed to be translated to a specification and test cases with criteria for the assessment of the test case.

Test case 1: Life operation. Roller failures may not stop the conveyor for long but they create several unwanted stops which could be avoided if the solution can predict such failures ahead of time, thus helping to make better planned stops that will not hamper the production negatively. The first test helps to assess the availability of the solution. The solution needs to be in life operation for at least three months. Within these three months, the availability of the application will be monitored, for the test to pass, there is no room for downtime should the model miss processing measurements from the data lake.

Assessment 1: The test was conducted during a validation period as an operational application. The application was fully operational and hosted on cloud servers. The nature of processing measurements are in batch and uses an intended delay while always backing up processed measurements. This set up brings some sort of protection against outages should either the data lake or cloud servers become inaccessible for some period of time. The system was operational during the whole validation period with automatic restarts upon unforeseen errors or outages.

Conclusion: Successfully completed.

Test case 2: Unwanted unclassified object on belt. Unwanted objects that could abruptly stop the conveyor need to be seen early so that the conveyor can be stopped and the objects removed. Different objects other than iron ore pellets will be placed on the belt at least five times. The solution must be able to at least detects 4 of these objects, thus giving an 80

Assessment 2: During the validation period, tests were performed by placing foreign objects on the belt. A total of 22 objects were placed on the belt. Out of these, 19 objects were successfully detected during the tests. Objects missed were darker in colour or thin in shape and blended in with the bulk material which may pose a difficulty since the solution is camera based. Naturally, the solution has successfully detected objects such as big rocks and lumps of material during the validation period. *Conclusion Successfully completed.*

Test case 3: False positive detection of objects on belt. To be able to perform this test, the solution must run for a year. During this time the solution will be monitored to see how many detections are made, and evaluated how many of these were false positive detections. The evaluation is done either by manually inspecting the frames together with operator personnel, or checking inspection logs. The test should pass with a false positive rate of two per week.

Assessment 3: For this end the belt monitoring system was used as a guidance which can generate less than two false positives per week in average. The solution has not ran for a year, thus validation period lasted for approximately six months. During this period of time the number of false positive detections were 962 frames in total. After re-tuning of the algorithm the false positive of the algorithm is down to 143 for the period. This is still more than the anticipated 52 false detections for a 6 month period. It needs to be noted that the data availability for learning true positives and false positives was very limited and longer operational periods are of help here. The tendency is well towards achieving the false positive rate. *Conclusion: Not completed yet.*

Test case 4: Unwanted large objects on belt. At the test site, unwanted large objects on the belt are rare but have the highest consequence level meaning they pose the risk of halting the whole operations when occurring. Like test case 02, all the video streams for the small and big objects placed on the conveyor belt will be saved for fine tuning of the algorithm.

Assessment 4: In test case 2 and 4, the sizes were split evenly among the 22 objects placed meaning 11 were large and 11 small. The system missed three out of these objects and the deciding factor was more the coloring and how slim the object appeared to be on the actual belt, rather than the actual size. Dark objects and very slim objects, for instance a brown cardboard cane blended in with the pellets. Nevertheless, the cane can be classified as a small object whereas the other two missed were large dark-colored cardboard cutouts. *Conclusion: Successfully completed.*

Test case 5: Prediction of component failure. This test case is connected to the components with priority 1, namely the roller components. In order to perform this test, data is aggregated from the data lake and run through the condition monitoring solution, and the indications for upcoming failures are collected. A comparison with maintenance and inspection protocols for the same time period is made. Then a classification of the predictions as true or false positive is made. This test has passed when 80% of the maintenance and inspection data are predicted in good time.

Assessment 5: In test case 5 there was a problem of not knowing the RUL of components beforehand. Maintenance logs only contained information about rollers being exchanged, but not specifically which roller. Usually in the test plant

there is approximately exchanges of 10% of the total rollers, which means that the rollers exhibit an operational life which exceeds the validation period. Moreover, maintenance data could not be used for validation purposes, which is important to consider in the lessons learned. This also meant that the amount of maintenance data can not be considered in quantification.

To assess the test case a specific experiment was conducted where two newly installed idler sets were raised vertically by 70mm, to speed up the degradation process due to increased IML which results in higher friction and downward force from the belt weight on the idler (CEMA, 1997). It was difficult to foresee if the degradation model would fit for such an abnormal geometric change and also what the life expectancy for a roller would be in real life. This modification was also reflected in the model to see if it could predict an early failure.

For the decision support on the prediction of an upcoming failure, thresholds for RUL and RoC were defined. The achievement of the threshold would then trigger a highlighting of the roller in the web app. The RoC was very high, and RUL was immediately reducing immensely and in real life the rollers failed very rapidly. One of the rollers failed already after one day. The decision support would indicate the rollers properly, but it is difficult to understand if the degradation model is capable to reflect such abnormal behaviour. *Conclusion: The test case needed to be modified. Modified test case completed successfully. Remaining question marks due to lack of realism in the modified test case.*

Test case 6: Increased availability. Preferably the value creation is quantified prior to commissioning, but in case of the proposed belt conveyor condition monitoring solution, an organizational change is needed to follow up the roller exchanges accurately. If the roller exchanges are not followed up in detail, the performance of the indications is negatively affected and the availability can not be quantified correctly. To conclude, this test case is not feasible and can not be performed. It was therefore decided to discontinue test case 6.

4. DEPLOYMENT AND OPERATION

The solution is a SaaS solution which means there is no software installation at the site. Instead, the solution needs to be integrated with the IT infrastructure of the asset, namely their data lake. The integration uses standard APIs. Furthermore, the asset site needs to make the SCADA, DCS and maintenance system data available in the data lake as specific tags that can be exported to the SaaS solution.

During inspection, maintenance and evaluation, the operator personnel is able to interact with the system through a web-app interface. After login, the user is presented with an interactive dashboard of the system as shown to the right in Fig. 3, and tables presenting components that are either deviating

or have low predicted remaining useful life. Both systems, namely the model for component condition are deployed separately and store results in the same manner. Nevertheless, they are presented as the same SaaS and can independently be shut off for modularity should future customers not be in need of either one. The app allows operators to interact with the model. It loads current condition data calculated by the degradation model from the database and presents in on the dashboard. Figure 2 shows the software architecture of the setup. The dashboard shows the belt conveyor, with sections coloured in a traffic light scheme depending on the status of components within section along with aforementioned lists ranking components by deviations or RUL.

In the side view bar as shown in Fig 3, forms are present for the user to fill out exchange or deviation data. The forms interact with the system in the manner to either reset RUL of components (exchanging components) or by listing them as deviations that need further inspection. Furthermore, detections from object detector are shown as a list the possibility for the user to mark detections as "False positive" should the detection not be correct. The forms are thus used as a feedback to the system with ground truth information.

4.1. Component condition monitoring

Since the digital twin of the plant system is run in parallel with the belt conveyor, there is no initialization criteria for the system, other than initial calibration described in section 2.1. Considering the conveyor contains rotating components in the thousands, information regarding the installation date and maintenance actions on specific components such as rollers are rather sparse. This was the case with the asset in question and consequently the rollers were initialized to 50% RUL, and were reset to 100% once maintenance actions are taken. The digital twin model is created in Python and deployed as a Docker container to a cloud cluster for operation. The model acts on incoming data, calculates the degradation of remaining useful for each affected rotating component and updates individual records for the components in a database. A rough software architecture for the model can be seen in Fig 1. The maintenance actions are noted through forms in the web app and is coupled to the database for the digital twin. Exchange actions resets the component on the DT level of the figure.

4.2. Edge solution for object detection

Contrary to the component condition monitoring, this solution suggests, hardware installations on the site. Namely the at least camera and the light sources with respectively cabling. The solution further suggests an edge server solution at the site but is not a limiting factor. For commissioning, the following step wise approach needs to be realised:

1. Select a strategic position of the installation of the camera and light source:

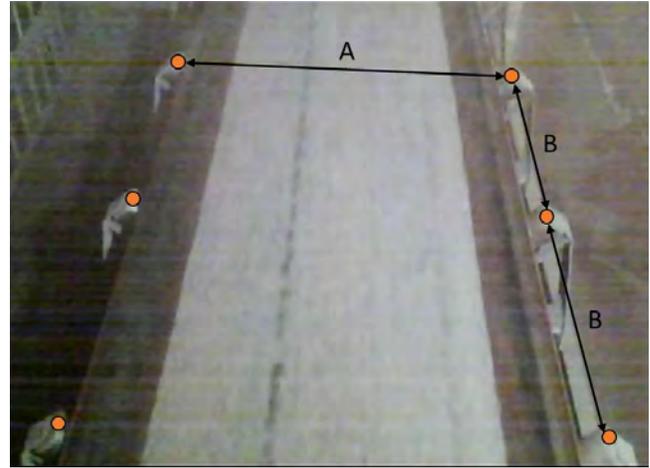


Figure 4. Drawing to indicate the needed measurements for the calibration.

- (a) Free field of view on the belt covering a length of 3.6m.
 - (b) No additional feeders downstream.
 - (c) Facing the incoming direction.
 - (d) Stable light conditions (limited number of commuting vehicles with head lights).
 - (e) Sufficient downstream distance to enable a full stop of the conveyor should dangerous objects be detected.
2. Installation of the camera and light sources.
 3. Enabling data stream, preferably RTSP and making the camera available on LAN.
 4. Installing edge server on site **if needed** and make it available on LAN and accessible remotely.
 5. Initiate operation of detection system for acquisition of training data. Exercise specific experiments with manually added objects for faster training and tuning.
 6. Proceed to validation and continuous operation for the solution.

Estimation of object sizes depends on the camera distance to the belt, its zoom and angle to the belt. Calibrating the size estimation requires some measurements on the conveyor at the installation site, as show in Fig4. The camera view of the belt should be so that the top three idlers are completely visible. These are indicated by the orange dots in the figure. Moreover, the measurements **A** and **B** are necessary and can be taken directly from construction drawings or measured on site.

5. EXPERIENCES

The test cases have been completed except for test case 3 since the application has not been in operation for the required time. The test case is not tied to any KPI but worth

mentioning is that tuning of the algorithm is an ongoing process to reduce the number of false positive detections and have a reliable as possible solution that is deemed trustworthy by the operator personnel.

The solution has been in life operation for more than 6 months. Naturally, cloud applications like any other require restarting when errors that were not intended for arise. Such errors may arise due to corrupt data or even power outages along the chain of the system architecture shown in Figs 1 and 2. Since the digital twin model of the plant always backs up streamed data from the asset data lake, it has the possibility of reprocessing measurements up until current time, the model does virtually not experience any down time. The object detector, even though it automatically restarts on errors and power outages, is affected by this since it streams live data. Nevertheless, the restarts require seconds of downtime and the same can be said for the web application. The system is judged to have well met the requirement of availability.

The system performs object detection of unwanted and unclassified with a high degree of precision. As mentioned in previous section, test cases 3 and 4 were conducted by placing objects on the belt. The detection rate of these tests were 86.3%, and beyond that, the system has made 240 detections of unwanted objects such as big lumps, rocks, slabs of concrete.

Prediction of component failure point in the right direction, but more validation is needed since the system has not tracked any component during its whole lifespan except for the two idler sets purposely installed with a vertical misalignment in test 5 to get a faster degradation rate. One roller failed at the 20-day mark and the other failed after only two days. The model predicted failures within 17 and 30 days respectively. The conclusion from these experiments is that since the model seems fairly accurate since it could predict such a large reduction in RUL from the new operating characteristics, although with some uncertainties for the experiment.

Regarding if the system provides increased availability, the project group is positive that this can be achieved using the system although not possible to establish quantifiable understanding during the project runtime. Current maintenance practices does not manage or track individual rollers/idlers and the maintenance system does not offer a means of managing them. As a result, downtime could not be properly established for either baseline and validation. Rollers being exchanged during the project were not reported as intended and benchmarking subsequently not possible.

5.1. Lessons learned

The project set out with the ambitious goal of raising the TRL of a prototype solution for condition monitoring of belt conveyor systems to 9 which is a market ready solution with vali-

dated performance characteristics. During the project, several challenges were encountered that were not foreseen and are obviously vital for later efforts. Those will be discussed in the following.

5.1.1. Maintenance practice in conflict with validation

Current maintenance practices might prevent the establishment of a baseline or the benchmarking of a new technology. It is therefore vital to assess current practices, data acquisition and data quality early on during the specification phase to make sure that there is sufficient high-quality data available for the test and validation strategies of the new technology. Furthermore, the validation should consider maintenance practices with respect to the degree of changes that are necessary to implement. Too many changes will induce delays and a higher amount of commitment and engagement of maintenance personnel and operators, which can conflict with the amount of time and resources planned in the project.

5.1.2. Covid-19 related delays and accessibility

Access to pilot and demonstration sites can be restricted and induce delays for the implementation, commissioning, and on-site activities. Validation plans need to consider a certain degree of flexibility and backup plans due to the currently ongoing pandemic. The complete project has been affected by this issue.

5.1.3. Real life ambient conditions

Real life ambient conditions may differ from the theoretical development of the solution and the foreseen real life behavior. Such factors are difficult or impossible to measure. These conditions might also occur during a limited time span and might be hard to foresee. It is important to collect as much information as possible during plant visits due build a good intuition. The validation can then use the collected knowledge and make decisions on abnormal behavior and/or the need to exclude them from validation.

5.2. Normal operation supersedes experimentation

Execution of planned tests in normal operation is difficult as modifications that affect operations impose a risk that could in the worst case render stops. Such as adding unwanted objects to a material stream. Normal operation will therefore supersede experimentation and tests cases can thus be delayed. While this is not unknown news, it may be overlooked during test case planning.

6. CONCLUSION

The assessment of the test cases performed shows promising performance characteristics for the system in using it as a decision and maintenance support tool. Used per guidelines,

the solution keeps track of and predicts the future status of individual components such as rollers and pulleys within a belt conveyor system. Moreover, the solution can detect unwanted objects on the belt, imposing a hazard for the belt life. The object detector shows high detection capabilities and relatively low false positive rates during the tests.

It further shows that the solutions can predict roller failures using given thresholds, and the object detector fulfills the true positive detection requirement. Principally, improvements are needed on the false positive rate which is exceeding a threshold taken from current performance of a belt monitoring system. As for the desired increase in availability, the lack of data needed to assess a baseline level and current maintenance practices leaves impossible to express or measure. In general, it can be concluded that the developed solution is commercially viable for implementation and value creation.

The authors thus recommend:

1. **Early on involvement from operators.** For testing and commissioning of the solution since their insights on smooth operation and possible earlier use of decision support tools aids in tailoring the UI/UX to the customer. This would also be a natural step in training and supporting different roles at the site.
2. **Object warning.** A lot of hauling sites have cameras that stream live data to operators in control rooms for manual inspection. These systems usually back up video during a set period of time. Generating a timestamp that could be sent via TCP from the object detector to this system for video replay in the control room can be used as an intermediate step towards automatic conveyor stops from a detection. It enables an automated labelling of frames from the action operators take (Stopping/not stopping belt).
3. **Hand-held devices.** The accuracy of reporting component exchanges is vital for the system performance. Using hand held devices in the field would enable the direct reporting of component exchanges without relying on secondary note-taking. It seems less likely to miss a report action and reduces effort.

Finally, it should also be noted that the excellent collaboration of the project participants during the given corona situation made the successful completion of the project possible. While less time was spent on travel to and from the site, it created an overhead in online meeting and more efforts by the local project participants.

ACKNOWLEDGMENT

Funding received under grant number 15099 from EIT Raw-Materials (in turn funded by the Framework program Horizon Europe of the European Commission) is hereby gratefully acknowledged. The authors would also like to thank

LKAB Narvik for the great support cooperation throughout the project.

REFERENCES

- Al-Kahwati, K., Saari, E., Birk, W., & Atta, K. (2021). Condition monitoring of rollers in belt conveyor systems. In *2021 5th international conference on control and fault-tolerant systems (systol)* (p. 341-347). doi: 10.1109/SysToI52990.2021.9595269
- CEMA. (1997). *Belt conveyors for bulk materials : Fifth edition*. Conveyor Equipment Manufacturers Association.
- Liu, X., Pang, Y., Lodewijks, G., & He, D. (2018). Experimental research on condition monitoring of belt conveyor idlers. *Measurement*, 127, 277–282.
- Liu, X., Pei, D., Lodewijks, G., & Zhao, Z. (2019). Intelligent maintenance of belt conveyors using machine learning. In *13th international conference on bulk materials storage, handling and transportation (icbmh 2019)* (pp. 234–243).
- Lodewijks, G. (2004). Strategies for automated maintenance of belt conveyor systems. *Bulk Solids Handling*, 24(1), 16–22.
- Lodewijks, G., Li, W., Pang, Y., & Jiang, X. (2016). An application of the iot in belt conveyor systems. In *International conference on internet and distributed computing systems* (pp. 340–351).
- Lodewijks, G., & Ottjes, J. A. (2005). Intelligent belt conveyor monitoring and control: theory and application. In *Beltcon 13 conference, fourways, south africa* (pp. 1–9).

BIOGRAPHIES



Kammal Al-Kahwati Received his M.Sc degree in Computer Science and Engineering from Luleå University of Technology in 2020. He has a background as a software engineer in the automotive industry and is currently working as an analytics and software engineer at Predge. His work include research topics surrounding condition monitoring and health prediction of rotating components and heavy machinery.



Wolfgang Birk is Head of Analytics at Predge AB and Professor of Automatic Control. He holds a M.Sc. degree in Electrical Engineering from University of Saarland, Germany (1997), a Ph.D. degree in Automatic Control from Luleå University of Technology (2002), and Professor of Automatic Control (2015). Birk has a background in the development of condition monitoring systems,

process control systems for resource efficiency as well as active safety systems in the automotive sector. His research work has led to control and monitoring solutions increasing the resource efficiency, utilization and availability for energy system, iron and steel making processes, processes in the pulp

and paper industry, and railway systems. In the railway sector, his main interest and expertise is the use of on-board, way-side, and track monitoring systems for condition monitoring in operation and maintenance.

A Case-study Led Investigation of Explainable AI (XAI) to Support Deployment of Prognostics in industry

Omnia Amin¹, Blair Brown², Bruce Stephen³ and Stephen McArthur⁴.

^{1,2,3,4} *Department of Electronic and Electrical Engineering, University of Strathclyde, 250 George St, Glasgow G1 1XQ*

omnia.amin@strath.ac.uk

Blair.Brown@strath.ac.uk

Bruce.Stephen@strath.ac.uk

S.mcarthur@strath.ac.uk

ABSTRACT

Civil nuclear generation plant must maximise its operational uptime in order to maintain its viability. With aging plant and heavily regulated operating constraints, monitoring is commonplace, but identifying health indicators to pre-empt disruptive faults is challenging owing to the volumes of data involved. Machine learning (ML) models are increasingly deployed in prognostics and health management (PHM) systems in various industrial applications, however, many of these are black box models that provide good performance but little or no insight into how predictions are reached. In nuclear generation, there is significant regulatory oversight and therefore a necessity to explain decisions based on outputs from predictive models. These explanations can then enable stakeholders to trust these outputs, satisfy regulatory bodies and subsequently make more effective operational decisions. How ML model outputs convey explanations to stakeholders is important, so these explanations must be in human (and technical domain related) understandable terms. Consequently, stakeholders can rapidly interpret, then trust predictions better, and will be able to act on them more effectively. The main contributions of this paper are: 1. introduce XAI into the PHM of industrial assets and provide a novel set of algorithms that translate the explanations produced by SHAP to text-based human-interpretable explanations; and, 2. consider the context of these explanations as intended for application to prognostics of critical assets in industrial applications. The use of XAI will not only help in understanding how these ML models work, but also describe the most important features contributing to predicted degradation of the nuclear generation asset.

1. INTRODUCTION

Although there are many different approaches in PHM, AI and ML powered techniques have recently seen a surge across applications in different industries. These Industries are continuously exploring AI and ML methods to ensure reliable and sustainable operations for their industrial assets. The goal of using these techniques is to carefully maintain industrial assets, to ensure that they fulfil their dedicated functions and also to avoid any unnecessary asset downtime. However, in industries where safety and reliability are crucial, the use of AI techniques impose a challenge of non-transparency to stakeholders. Stakeholders need to understand how ML techniques work and how they produce their outputs in order to build trust in decisions based upon these outputs and realise AI/ML deployments within their industries. Explainable AI (XAI) helps in explaining these techniques and make it more transparent to stakeholders. XAI has a vital role in PHM systems as it helps nurture confidence in AI techniques used while the function and performance of the underpinning AI systems and the associated asset remain intact. This paper illustrates the need for XAI in PHM and how XAI can help non-ML experts adopt ML models through demonstration on diagnostic and anomaly detection case studies. This paper proposes novel algorithms that will help non-ML experts to understand the explanation produced by XAI tools. The goal is to give the reader an insight into the importance of combining XAI and PHM. This paper is organized as follows: Section 2 states the problem and proposes a solution, Section 3 describes the different approaches that can achieve explainability, Section 4 demonstrates the proposed approach used for this paper, Section 5 explains the algorithms developed, Section 6 introduces three different case studies in which the proposed approach has been applied, Section 7 discusses the solution proposed and finally section 8 summarises conclusions and draws directions for future work.

Omnia Amin et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. PROBLEM STATEMENT

Many AI and ML diagnostic and predictive applications use black-box type models because their outputs provide better performance than simpler, and therefore more transparent, white-box approaches. However, stakeholders in regulated industries such as Nuclear, base their operational decisions on understanding how models generate their predictions (Preece, Harborne, Braines, Tomsett, & Chakraborty, 2018). In the case of civil nuclear generation, a fault prognostic model may invoke a decision to take the station offline while the fault is investigated, which in turn will incur maintenance costs and lost generation revenues. XAI helps users understand the underlying structure of black-box machine learning models and how they produce their outputs; hence, boosting user's confidence in these models and encouraging them to use them. Unfortunately, most XAI that are in use produce explanations in a technical format that is not easily understandable to a non-ML expert (Bove, Aigrain, Lesot, Tijus, & Detyniecki, 2022), which in the case of power generation, most operational staff will be. Research shows that experts in the application domain tend to trust machine learning models when they are provided with human-friendly explanations that will enable them to understand the rationale of ML models (Bove et al., 2022). Also, there is a requirement for distinctly different explanations for stakeholders in different application domains (Mohseni, Zarei, & Ragan, 2018). To pursue this challenge, this paper proposes a novel application of a set of algorithms that translates explanations generated from XAI tools into human understandable text-based explanations.

3. DIFFERENT APPROACHES TO XAI

Explainability (Interpretability) (Carvalho, Pereira, & Cardoso, 2019) can be achieved through different approaches and they can be classified according to different criteria. In this section, we will explore some of the well-known classifications (Barredo Arrieta et al., 2020):

3.1. Pre-model, During Model and Post-Model

Explainability can be achieved through different complementary approaches. One of these approaches depends on when XAI techniques are applied. They can be applied as: 'Pre-Model', 'During model' and 'Post-model methods' (Stiglic et al., 2020) (Carvalho et al., 2019). 'Pre-model' is done in the first stage of model development after obtaining the related data and before selecting the desired ML model appropriate for the problem statement. The primary goal of using pre-model methods is to understand and describe the data used in ML model and how the data health and structure influence the model. 'During Model' is an approach to ensure explainability through the use of transparent models, which are models that are inherently understandable for humans. (Doran, Schulz, & Besold, 2017) Using transparent models is one ap-

proach to achieve interpretability. In these models, humans can easily understand how inputs are mathematically mapped to outputs by having technical knowledge of the model itself and the algorithms used in the models (Molnar, 2020). One drawback of this approach that it is model-specific, and the model design process is limited by the number of representative models available to choose from. Interpretable models include linear regression, logistic regression, generalized linear models, and decision trees (Molnar, 2020). Finally, 'Post model' or 'Post-hoc Methods' is an approach applied after choosing the ML model and after obtaining predictions from these models. Currently, most black-box models are explained using a post-hoc approach. This approach is used for complex models in which humans cannot understand the underlying decision-making mechanism. The advantage of post-hoc approaches is that they do not affect the performance of a complex model as it treats the model as a black-box (Dosilovic, Bri, & Hlupic, 2018). Post-hoc approaches can be primarily classified into three groups:

1. Gradient based attribution methods such as saliency maps (Simonyan, Vedaldi, & Zisserman, 2014) which assign importance scores to each input feature and show which parts of the input are most important.
2. Surrogate Models such as MUSE (Model Understanding through Subspace Explanations) (Lakkaraju, Kamar, Caruana, & Leskovec, 2019). In this approach, black-box models' behavior is explained in sub-spaces defined by specific features that are of user interest.
3. Post-hoc approaches via perturbation: This approach uses perturbations of the input data to generate pairs of inputs and outputs, then uses simple models e.g., linear models to explain the prediction obtained. Examples of techniques that use this approach are LIME and SHAP tools. Shapley Additive exPlanations (SHAP) tool computes feature importance by computing the contribution of each feature to the output obtained. These contributions are calculated using coalitional game theory, where features represent players in a coalition (Molnar, 2020). SHAP tool increases transparency by producing SHAP values for each instance in the data set (Molnar, 2020). SHAP values can be aggregated to provide global interpretability for machine learning models. It is considered an optimal approach for providing interpretability since it is built on a solid theory (Lundberg & Lee, 2017). Some advantages of SHAP is that it is based on a solid theoretical theory and it can provide local and global explainability by providing SHAP force plot for local explainability and SHAP summary plot for global explainability as shown in figure (1) (Lundberg & Lee, 2017). Due to the benefits and wider adoption of SHAP by the AI community, application and development of SHAP values will be a focus of this paper.

3.2. Global and Local explainability

A second approach to classifying interpretability methods is according to the scope of how they assess the underlying model, i.e., from a global or local perspective (hui Li et al., 2022) (Bhatt et al., 2020) (Angelov, Soares, Jiang, Arnold, & Atkinson, 2021). Global interpretability: Global methods help users to logically understand the relationship between all input variables and the predicted output. They help in forming an overall understanding of the behavior of the model (Doran et al., 2017). Users are able to understand all the different possible outcomes. In contrast, local interpretability provides an explanation for one instance or region of the modeled space, and the associated contribution of that instance or space to the overall output (Bhatt et al., 2020).

3.3. Model-Agnostic and Model-Specific explainability:

In model-agnostic techniques, there is the flexibility to choose any machine learning approach. The machine learning model in a model agnostic approach is treated as a black-box by separating the explanations from the model, thus giving the flexibility to choose any ML model, alongside any representation and explanation (Molnar, 2020) (Angelov et al., 2021). One disadvantage of using model-agnostic techniques is the possibility of having inconsistent local explanations (Ribeiro, Singh, & Guestrin, 2016) (Ribeiro et al., 2016). In model specific techniques, choice is limited to specific models because methods are based on the internal workings of specific models, and it is hence difficult to change to another model (Molnar, 2020).

4. NEW PROPOSED APPROACH TO XAI

The goal is to develop various options for extracting explainability (interpretability) from predictive or diagnostic analytic tools. These extracted explanations being required to be presented to decision-making stakeholders who are non-ML experts in human-friendly context. To achieve this goal, four complementary explanatory stages have been identified (see figure (2)). Each stage is presented next with more details:

1. Data pre-processing: The first stage eases the understanding of the data set used and recognizes the features contained therein. The quality of data is assessed and transformed into an understandable format that can be used later in ML/analytic models.
2. Prediction Models: The second stage is to choose appropriate machine learning prediction model(s). Most ML models are considered black-box models, in which we cannot understand how these models work and how inputs are mapped into outputs. Therefore, there is a need to develop and deploy XAI techniques that generate explanations on predictions made, enabling industry stakeholders to understand the machine learning models adopted.

3. Applying XAI tools: Applying XAI tools to provide understandability of how ML models work and why they produce these predictions. For this paper, a widely adopted post-hoc XAI tool is used. SHAP is applied to provide local and global explanations. SHAP generates more reliable explanations than other XAI tools, and it can provide local explanations for a single predication (e.g. why a specific prediction has been made , what are the most important features contributing to this prediction, and the impact of each feature on the prediction) and a global explanation to provide a holistic understanding of how the ML model works. However, SHAP produces these explanations in the form of complex plots which are not easy to understand, especially for a non-ML expert. This is the rationale for introducing a final stage to translate these explanations into a more understandable format.
4. Generating human understandable explanations: How to communicate explanations to non-ML-experts in the application domain is important. SHAP plots are not always easy to understand, even for a data scientist. This fact leads to the need to translate these plots into a human-understandable context that will result in bridging the gap between ML experts and stakeholders. In this stage, a set of novel algorithms have been developed to translate SHAP local and global explanations to generate human understandable text-based explanations.

5. AUTOMATED HUMAN-UNDERSTANDABLE-TEXT GENERATION ALGORITHMS

In this section, the algorithms used to translate complex SHAP plots to human-understandable text based explanations are described.

5.1. Translating SHAP local explanation plots

After applying the SHAP tool in order to provide interpretability, the resulting explanations are produced in the form of complex plots. This paper demonstrates a novel approach, where these plots are translated into text-based explanations.

For SHAP local interpretability, a SHAP force plot is produced for a single prediction, providing the most important features, and the impact of each feature on the output at a local level. If a feature has a positive SHAP value, this indicates that the feature value has a positive impact on the prediction. However, if it has a negative value, this indicates that the feature has a negative impact on the prediction and finally, if the SHAP value equals zero, then this feature has no impact on the output.

How this information is translated into human-understandable text is demonstrated in Figure (3), where a flow chart explains how the logic behind the code that has

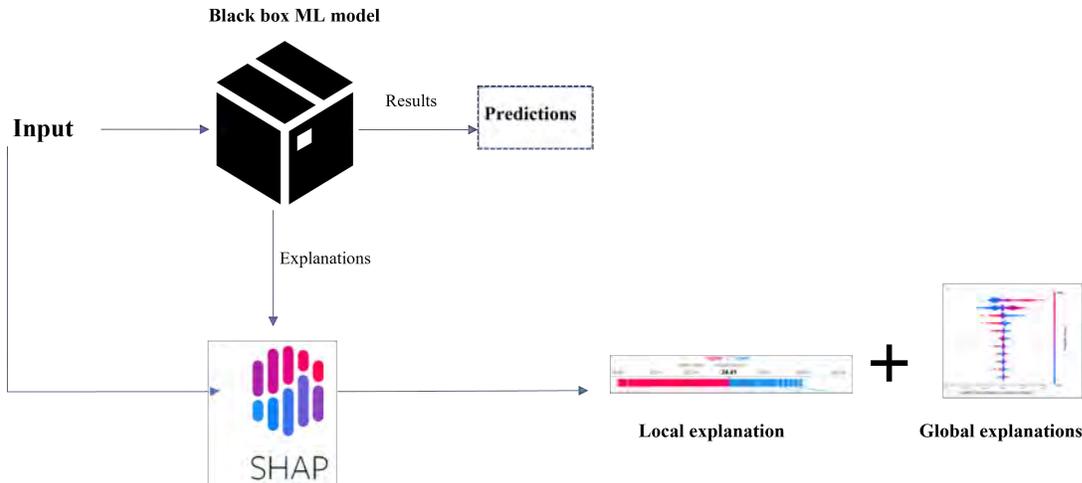


Figure 1. SHAP local and global explanations.

been developed for the automated text-generating process.

5.2. Translating SHAP global explainability plot

For Global explainability, each feature has its SHAP values. The SHAP values for each feature are aggregated and then compared. After that, all the features are ordered according to their aggregated SHAP values. Features with higher aggregated SHAP values have greater impact on the output and vice versa. How this information is translated into human understandable text is demonstrated in Figure (4), where a flow chart explains how the logic behind the code that has been produced for the automated text-generating process.

6. CASE STUDIES

The novel approach to human-understandable XAI, described in Sections 4 and 5, has been applied to three different case studies, as follows.

6.1. Case study 1 : Combined Cycle Gas Turbine (CCGT)

In the first case study, a publicly available data set consisting of operational measurements from a Combined Cycle Gas Turbine (CCGT) generator has been used. An open-source data set has been chosen as the first case study to facilitate easy application of XAI tools and also to make the work reproducible. The CCGT data set was curated over 6 years (2006-2011) and has been previously used to show machine learning models for predicting power output based on environmental conditions (Wood, 2020) (Tüfekci, 2014)

6.1.1. Data pre-processing

The data set composes the following operational measurements from the turbine, generator, and control valves:

1. Ambient pressure (AP).

2. Exhaust Vacuum (V).
3. Ambient temperature (AT).
4. Relative humidity (RH).

These parameters are used to predict the net hourly electrical energy output (PE) of the plant. In this case study, it was shown that the relationship between environmental conditions and power output could be clearly identified and explained. In figure (5), some statistical properties about CCGT data set are provided.

6.1.2. Modelling

Three different candidate models have been implemented for the explainability case-study: linear regression, random forest and XGboost. These ML models were used to predict the output power and their performances were compared using three performance metrics that are usually used to compare performance between different regression models: Root mean squared error (RMSE) metric, which measures the average error performed by the model, R2 score which specifies how close the calculated values are plotted to the actual data values and Mean squared error (MSE). The metrics for Gradient Boosting Regressor showed improvements over the Linear Regression Model and the random regression model. There are 3 key performance metrics (See table 1) used to assess how well each model is performing. After evaluating all the models, XGBoost Regression Algorithm was found to give the best performance with R-squared = 0.97 and RMSE = 3.069.

6.1.3. XAI Application

In this stage, SHAP has been adopted to provide explanations to the ML predictions. It generates explanations in a form of visualizations that are quite complex and are not always intuitive. In figure (6), SHAP summary plot produced using the

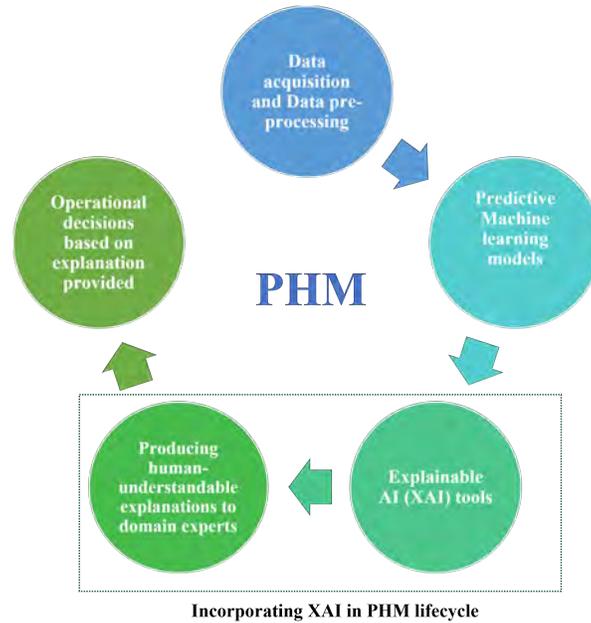


Figure 2. Proposed approach.

Table 1. Comparison of ML models performances.

ML models	R2 Score	MSE	RMSE
Linear regression	0.92	20.637	4.543
Random forest	0.94	15.278	3.909
XGBoost	0.97	9.419	3.069

XGBOOST model. According to SHAP summary plot shown in figure (6), the most important features in descending order are : Temp, Vacuum, Pressure and finally Humidity. The impact of each feature is also shown (e.g. high values (shown in red color) of Temp has a negative impact on the output power causing the output power to decrease while low values (shown in blue) of Temp has positive impact on the output power causing the output power to increase). In the SHAP force plot (SHAP local explainability plot), shown in figure (7), features like Temp, vacuum and pressure (shown in red) causing an increase in the predicted output power. The visual size for each feature in SHAP force plot (size of the arrow) shows the magnitude of each feature’s impact. According to this local explanation the most important features in descending order are Temp, Vacuum, Pressure and finally Humidity.

6.1.4. Generation of human-understandable explanations

As described in Section 5, a set of algorithms have been developed to achieve the task of translating SHAP plots into text-based explanations for ease of comprehension. In figure (8), an example of the text-based explanations generated by translating the SHAP local explainability plot (SHAP force plot) is shown in figure (7). The text-based explana-

tion clearly describes the most important features of the plot and the associated impact from each feature value contributing towards the predicted output power (e.g. Temp =11.37 is considered a low value after comparing it to the mean value of Temperature in the data set and has a positive impact on the output power pushing the output power value higher). Figure(9) shows the automated text-based explanations for the SHAP summary plot shown in figure(6). While Figure (10) shows summary statistics for each feature, including: the number of values for each feature that have no impact on the output power; the number of values that are considered high and have high/low positive impact on the output pushing the output power to increase; and the number of values for each feature that are considered low and have a high/low negative impact on the output power, causing the prediction to decrease.

6.2. Case study 2: Boiler Feed Pump Gearbox Data set

A gearbox is a mechanical device used to increase or decrease the speed of another part connected to it along a rotating drive-train. The objective of this case study is to apply XAI tools to provide explanations to predictive models applied to a gearbox data set related to a boiler feed pump and then ease the ability to understand these explanations through the application of the auto-generated novel text-based algorithms proposed in this paper. The modelling aim was to investigate how the controlling stop-valve position values affect rms-vibration and the operational consequences associated with increased stop-valve position. Increased vibration in boiler feed pump lead to decrease in the performance of

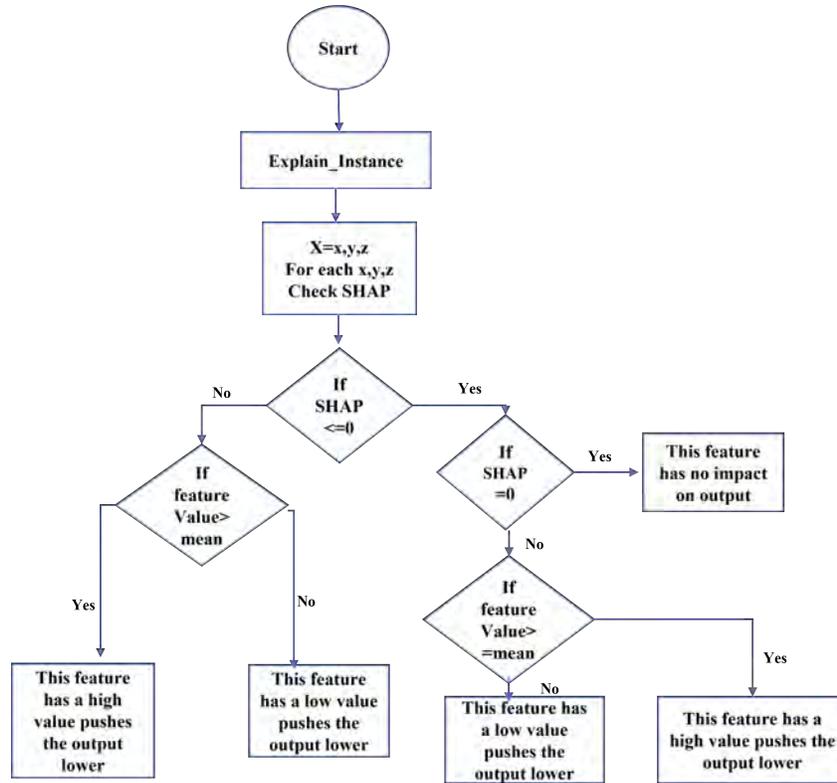


Figure 3. Flowchart to translate SHAP force plot.

pump and result in damage to some pump parts. In this case study, it was shown that the relationship between stop-valve position and rms-vibration could be clearly identified and explained.

6.2.1. Data pre-processing

The data set used comprises of different valve positions and rms-vibration to investigate if there is a correlation between the valve position and rms-Vibration. This data was collated and provided by a real operational boiler feed pump in the power generation industry. The following operational measurements were used to create the predictive model:

1. Stop-valve position.
2. Rms-vibration.

Before machine learning prediction models can be used, the time-series data set has been re-framed as a supervised learning problem, resulting in a sequence of input and output pairs. Reframing the data set removes the complexities around the prediction problem and can give more reliable forecasts. After re-framing the data set to a supervised learning problem, the following operational measurements used to predict rms-vibration(t+1) are: stop-valve-position(t-2), rms-vibration(t-2), stop-valve-position(t-1), rms-vibration(t-1), stop-valve-position(t), rms-vibration(t) and stop-valve-position(t+1).

Table 2. Comparison of ML models performances.

ML models	R2 Score	MSE	RMSE
Linear regression	0.96	0.001417	0.03766
Random forest	0.96	0.00144	0.0379
XGBoost	0.95	0.00151	0.03889
Ensemble Model	0.96	0.001455	0.03815

6.2.2. Modelling

Similar to the previous case study, the same three different ML models were assessed for their effectiveness: Linear regression, Random Forest, and XGBoost - all being used to predict rms-vibration(t+1). These ML models were then combined using an averaging ensemble model to improve the overall performance. Performances have been compared as seen in table (2) using three different performance metrics. As shown in table (2), Ensemble model has not improved the overall performance and linear regression has the best performance of the all models. It is concluded from these results that a linear regression model should be selected to create the SHAP values.

6.2.3. XAI application

This case study adopted SHAP to provide explanations for the ML predictions. Figure (11) is the SHAP summary plot

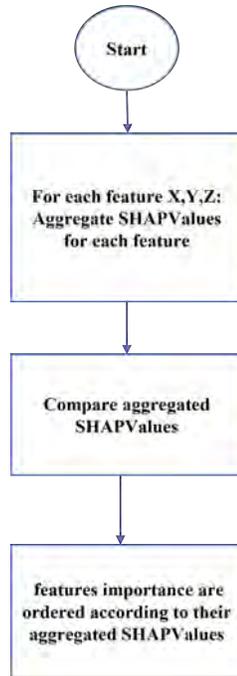


Figure 4. Flowchart to translate SHAP Summary plot.

	Temp	Vacuum	Pressure	Humidity	Power
count	9568.000000	9568.000000	9568.000000	9568.000000	9568.000000
mean	19.651231	54.305804	1013.259078	73.308978	454.365009
std	7.452473	12.707893	5.938784	14.600269	17.066995
min	1.810000	25.360000	992.890000	25.560000	420.260000
25%	13.510000	41.740000	1009.100000	63.327500	439.750000
50%	20.345000	52.080000	1012.940000	74.975000	451.550000
75%	25.720000	66.540000	1017.260000	84.830000	468.430000
max	37.110000	81.560000	1033.300000	100.160000	495.760000

Figure 5. Statistical details about CCGT data set.

using linear regression. The most important features are shown, with rms-vibration(t) being the most important feature and rms-vibration(t-2) the least important. The SHAP summary plot also shows the correlation between each feature and the output (e.g. rms-vibration(t) is positively correlated with the output, the higher the rms-vibration(t) is, the higher the output and similarly for rms-vibration(t-1)). In the SHAP force plot (local explainability plot), shown in figure (12), the most important feature for this prediction is shown in red: rms-vibration(t), having a positive correlation with the output (causing the output to increase). On the other hand, rms-vibration(t-1) shown in blue has a negative correlation with the output, causing the output to decrease.

6.2.4. Generation of human-understandable explanations

The techniques from Section 5 were then used to generate automated-text-based explanations that are easy to understand. In figure (13), an example of the text-based explanation generated corresponding to the SHAP force plot (local explanations plot) produced in figure (12). In figure (13), the most important features affecting the output for a specific instance are listed. Also, the impact of each feature value and whether this feature value pushes the output value higher/lower is shown (i.e. rms-vibration(t) has a low value for this instance that pushes the output higher). In Figure (14), a text-based explanation corresponding to SHAP summary plot shown in figure (11) is provided, denoting the most important features globally for the prediction model. Figure (15) shows summary statistics for some of the features, including: the number of values for each feature that have no impact on the output, number of values that are considered high and have high/low positive impact on the output pushing the output to increase and the number of values for each feature that are considered low and have a high/low negative impact on the output causing the prediction to decrease.

6.3. Case study 3: Thrust bearing wear predictive model

In this case study real condition monitoring data from feed-water pumps has been used to anticipate thrust bearing wear (denoted "median-TB") given operating parameters such as flow ("mean-Flow") and head ("mean-Head"). The data set used comprises of different values of flow and head. The data set is used to predict thrust bearing wear.

6.3.1. Data pre-processing

The data set used comprises of different values of flow and head. This data set is used to predict thrust bearing wear. Similar to the pre-processing stage for case-study two described in section 6.2 the time series data has been re-framed to a supervised learning problem from a sequence to pairs of input and output sequences. The following operational measurements are used to predict median-TB(t+1): mean-Flow(t-2), mean-Head(t-2), median-TB(t-2), mean-Flow(t-1), mean-Head(t-1), median-TB(t-1), mean-Flow(t), mean-Head(t), median-TB(t), mean-Flow(t+1) and mean-Head(t+1).

6.3.2. Modelling

Similar to the previous case-studies, the same three ML models are assessed for their predictive accuracy: linear regression, Random Forest, XGBoost have been used to predict thrust bearing wear (median-TB(t+1)). Then, ML models have been combined using the same averaging ensemble model to investigate whether or not the overall performance will be improved which in this case it didn't. Performances

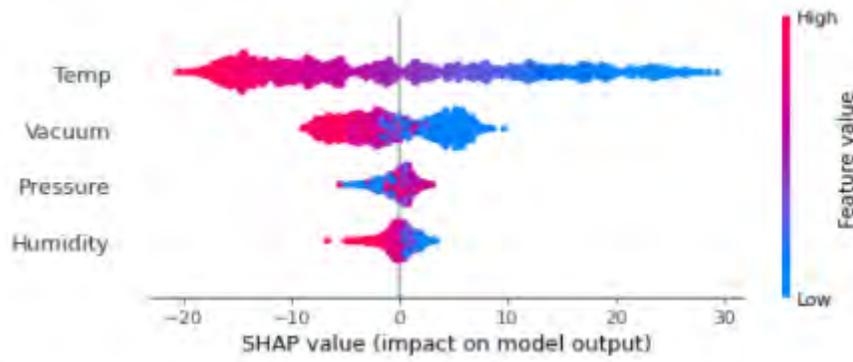


Figure 6. SHAP summary plot for case study 1.

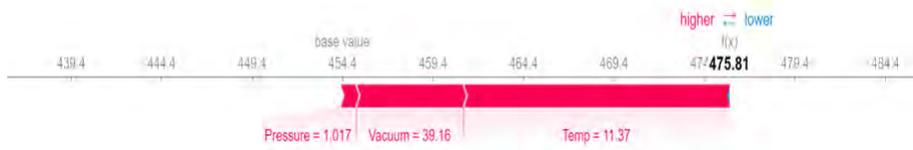


Figure 7. SHAP force plot for case study 1.

```

Features affecting the output power in order are
Temp
Vacuum
Pressure
Humidity
Impact of each feature value on output
Temp 11.37 Low value pushes the output power higher
Vacuum 39.16 Low value pushes the output power higher
Pressure 1016.54 High value pushes the output power higher
Humidity 87.05 High value pushes the output power lower
    
```

Figure 8. Text-based explanations for SHAP force plot in case-study 1

```

Most influential features are
Temp
Vacuum
Pressure
Humidity
    
```

Figure 9. Text-based explanations for SHAP summary plot in case-study 1

```

Global interpretability for Temp
No of points that have no impact on output = % 0.0
% of points that have low values and positive impact on output = % 45.03657262277952
% of points that have high values and positive impact on output = % 0.41797283176593525
% of points that have low values and Negative impact on output = % 1.4629049111807733
% of points that have high values and Negative impact on output = % 53.08254963427377
Global interpretability for Vacuum
No of points that have no impact on output = % 0.0
% of points that have low values and positive impact on output = % 44.40961337513062
% of points that have high values and positive impact on output = % 0.8881922675026124
% of points that have low values and Negative impact on output = % 6.948798328108673
% of points that have high values and Negative impact on output = % 47.7533960292581
Global interpretability for Pressure
No of points that have no impact on output = % 0.0
% of points that have low values and positive impact on output = % 21.26436781609195
% of points that have high values and positive impact on output = % 26.854754440961337
% of points that have low values and Negative impact on output = % 31.765935214211076
% of points that have high values and Negative impact on output = % 20.1114942528735632
Global interpretability for Humidity
No of points that have no impact on output = % 0.0
% of points that have low values and positive impact on output = % 43.155694879832815
% of points that have high values and positive impact on output = % 9.770114942528735
% of points that have low values and Negative impact on output = % 2.507836990595611
% of points that have high values and Negative impact on output = % 44.56635318704284
    
```

Figure 10. Text-based explanations representing simple statistics for SHAP summary plot in case-study 1

are compared as shown in table (3). Linear regression model has the best performance with the least mean squared error (MSE).

6.3.3. XAI application

Applying SHAP techniques to provide explanations to machine learning predictions produced the following results. Figure (16) depicts the SHAP summary plot, showing the

most important features contributing to model predictions. The plot shows SHAP values for each feature and the impact these features have on the model predictions. The most important features for this model from the global explanation perspective in descending order are: mean-Flow(t+1), median-TB(t), mean-Head(t+1), ..., and lastly mean-Head(t-1) as depicted in figure (16). From the SHAP summary plot mean-Flow(t+1) is positively correlated to the output, the

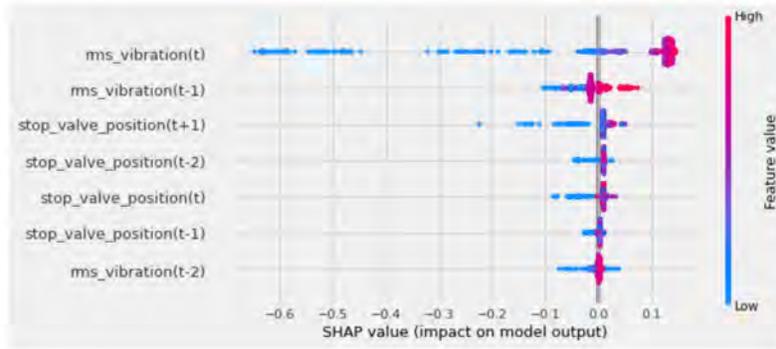


Figure 11. SHAP summary plot for case study 2



Figure 12. SHAP force plot for case study 2

Table 3. Comparison of ML models performances.

ML models	R2 Score	MSE	RMSE
Linear regression	0.9978	0.0000035	0.002
Random forest	0.997	0.0000042	0.00207
XGBoost	0.997	0.0000045	0.0021
Ensemble model	0.997	0.0000041	0.0021

higher the mean-Flow(t+1), the higher the output. Figure (17) shows the local explanation generated by SHAP for a single prediction using the linear regression model. In the SHAP force plot figure (17), the most important feature for this prediction (shown in red) is mean-Flow(t+1), having a positive correlation with the output (causing the output to increase). The second most important feature is median-TB(t) (shown in red), having a positive correlation with the output (causing the output to increase for this prediction).

6.3.4. Generation of human-understandable explanations

The techniques from Section 5 were used to generate automated-text-based explanations that are easy to understand. In figure (18), an example of text-based generated corresponding to SHAP local explanations plot produced above and shown in figure (17). In Figure (19), text-based explanations corresponding to the SHAP summary plot shown in figure (16), showing the most important features globally for the prediction model.

7. DISCUSSION

The aim of this work is to introduce XAI techniques into PHM systems. In this paper, a new approach has been proposed to produce a human-understandable format of SHAP produced explanations. Compared to other related literature which lacks human understandability, this approach makes it easier for non-ML experts to understand the results from explainability tools. The authors propose that the text-based representation of the SHAP process is easier and more intuitive to interpret because they allow non-ML experts to understand and engage with how ML models work. These text-based explanations will enable stakeholders to understand the impact of each input and the operational consequences associated with different inputs/values. The proposed approach has been used in three different case studies and demonstrates the provision of a human-friendly form of explanations to non-ML experts.

8. CONCLUSIONS AND FUTURE WORK

Exploiting the application of XAI tools in PHM can lead to increased confidence in PHM systems, encourage their adoption, and ultimately meet the assurances and quality required for PHM system deployment in safety-critical industries such as nuclear. In this paper, through the development and demonstration of a novel approach to the interpretation of a well-known post-hoc XAI technique (SHAP), it has been shown that explanations in a ‘human-friendly’ format can aid stakeholders (who are not necessarily ML experts) to rapidly interpret the technical explanations provided

```

instance no 0 taken at time: 2014-09-20 17:10:00
Turbine speed at this time:
TurbinespeedA= 4517.489258
TurbinespeedB= 4521.881348
features affecting the rms-vibration in order are
rms_vibration(t)
rms_vibration(t-1)
stop_valve_position(t+1)
rms_vibration(t-2)
stop_valve_position(t-2)
stop_valve_position(t)
stop_valve_position(t-1)
impact of each feature value on output
stop_valve_position(t-2) 100.8535309 High value pushes rms-vibration higher
rms_vibration(t-2) 0.8802593 Low value pushes rms-vibration lower
stop_valve_position(t-1) 100.8539429 High value pushes rms-vibration higher
rms_vibration(t-1) 0.8802593 Low value pushes rms-vibration lower
stop_valve_position(t) 100.8543549 High value pushes rms-vibration lower
rms_vibration(t) 0.8802593 Low value pushes rms-vibration higher
stop_valve_position(t+1) 100.8547668 High value pushes rms-vibration higher
    
```

Figure 13. Text-based explanation for SHAP force plot for case study 2.

```

Most influential features are
rms_vibration(t)
rms_vibration(t-1)
stop_valve_position(t+1)
stop_valve_position(t-2)
stop_valve_position(t)
stop_valve_position(t-1)
    
```

Figure 14. Text-based explanations for SHAP summary plot for case study 2

by black-box ML models that may comprise a PHM methodology. This subsequently increases their confidence in adopting the model that may have produced new prognostic insight for operational decisions. This new approach is intended to support end-users (who are not ML experts) interpret the outputs from SHAP and this benefit is realised through the creation of new algorithms that auto-generate text-based explanations based on SHAP summary and force plot outputs. These text-based explanations provide a more intuitive means of interpreting SHAP outputs, which are more generally intended for data scientists or other practitioners familiar with the field of study. The approach developed has been applied to three case-studies – two (2 and 3) of which are based upon operational data from a nuclear power station. They demonstrate that it is possible to produce more intuitive explanations than the standard graphical outputs produced by SHAP tools. These more intuitive text-based explanations can henceforth be more easily understood by the end-user of the related PHM algorithms, who may be unfamiliar with both: the ML predictive algorithm in its own right but also the methodology and format associated with SHAP. In addition, during the investigation associated with this paper, the authors have identified

```

Global interpretability for rms_vibration(t-2)
No of points that have no impact on output= % 0.0
% of points that have low values and positive impact on output = % 0.8630648997621474
% of points that have high values and positive impact on output = % 81.97757390417941
% of points that have low values and Negative impact on output = % 16.71083927964662
% of points that have high values and Negative impact on output = % 0.44852191641182465
Global interpretability for stop_valve_position(t-1)
No of points that have no impact on output= % 0.0
% of points that have low values and positive impact on output = % 3.9755351681957185
% of points that have high values and positive impact on output = % 0.12232415902140673
% of points that have low values and Negative impact on output = % 0.0
% of points that have high values and Negative impact on output = % 95.90214067278288
Global interpretability for rms_vibration(t-1)
No of points that have no impact on output= % 0.0
% of points that have low values and positive impact on output = % 0.12232415902140673
% of points that have high values and positive impact on output = % 81.14169215086646
% of points that have low values and Negative impact on output = % 17.44478423377506
% of points that have high values and Negative impact on output = % 1.291199456337071
Global interpretability for stop_valve_position(t)
No of points that have no impact on output= % 0.0
% of points that have low values and positive impact on output = % 0.81359157322460074
% of points that have high values and positive impact on output = % 83.7037037037037
% of points that have low values and Negative impact on output = % 3.961943594971118
% of points that have high values and Negative impact on output = % 12.320761128100578
Global interpretability for rms_vibration(t)
    
```

Figure 15. Text-based explanations for some of the features in case study 2 .

some limitations of the proposed approach that can be further improved to produce more robust and reliable explanations, and which are the focus on on-going work. One limitation identified, and associated with using a correlating post-hoc tool such as SHAP, is the absence of the ability to causally link the correlations identified by SHAP to related physical phenomena. Introducing causality into post-hoc XAI tools will help in providing more reliable explanations, both by related the correlations to the underpinning physics but also by potentially providing explanations in specific engineering domain contexts. In addition to these causality investigations, the authors have a further aim to develop additional/improved means of intuitively representing and subsequently interrogating AI explanations. Building on the content of the work described in this paper, the authors are currently developing techniques to auto-generate graph-based representations of the semantic knowledge embedded within AI explanations. The intended methodology aims to continue improving on

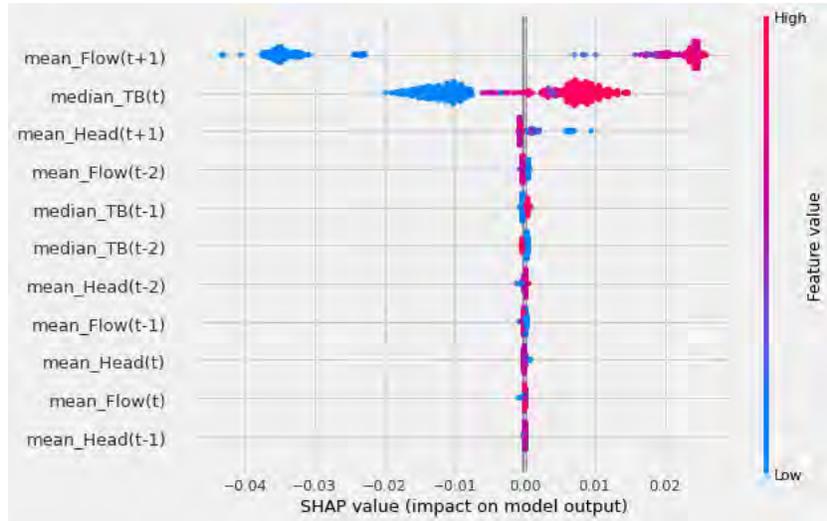


Figure 16. SHAP summary plot for case study 3.



Figure 17. SHAP force plot for case study 3.

```

instance no 0 taken at time: 30/10/2015 14:04
features affecting the median_TB(t+1) in order are
mean_Flow(t+1)
median_TB(t)
mean_Head(t+1)
mean_Flow(t-2)
median_TB(t-2)
median_TB(t-1)
mean_Head(t-2)
mean_Flow(t-1)
mean_Head(t)
mean_Flow(t)
mean_Head(t-1)
impact of each feature value on output
mean_Flow(t-2) 479.3611552 High value pushes median_TB(t+1) lower
mean_Head(t-2) 193.4650223 High value pushes median_TB(t+1) higher
median_TB(t-2) 0.001806837 High value pushes median_TB(t+1) lower
mean_Flow(t-1) 478.8041306 High value pushes median_TB(t+1) lower
mean_Head(t-1) 193.0229993 High value pushes median_TB(t+1) higher
median_TB(t-1) -3.11e-06 High value pushes median_TB(t+1) higher
mean_Flow(t) 477.8979441 High value pushes median_TB(t+1) higher
mean_Head(t) 192.6907872 Low value pushes median_TB(t+1) lower
median_TB(t) -0.000458766 High value pushes median_TB(t+1) higher
mean_Flow(t+1) 479.9311702 High value pushes median_TB(t+1) higher
mean_Head(t+1) 193.4374611 High value pushes median_TB(t+1) lower
    
```

Figure 18. Text-based explanation for shap force plot for case study 3

```

Most influential features are
mean_Flow(t+1)
median_TB(t)
mean_Head(t+1)
mean_Flow(t-2)
median_TB(t-1)
median_TB(t-2)
mean_Head(t-2)
mean_Flow(t-1)
mean_Head(t)
mean_Flow(t)
    
```

Figure 19. Text-based explanation for shap summary plot for case study 3

how (non-ML expert) end-users can adopt PHM through explanation of the related AI technique but in parallel also facilitate machine interactions and interfacing with the software-based explanation process. It is proposed that providing a means of machine interface to the explanation process can lead to the inclusion of techniques such as query language and more sophisticated graph manipulation; ultimately resulting in more insight and knowledge discovery, both for nuclear engineers hoping to adopt ML-based PHM techniques

but also more generally in industries with a similar deficit of ML capability.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support received from the UK's National Physical laboratory (NPL) and from the Advanced nuclear research centre (ANRC) at the University of Strathclyde, Glasgow.

REFERENCES

- Angelov, P., Soares, E., Jiang, R., Arnold, N., & Atkinson, P. (2021, 09). Explainable artificial intelligence: an analytical review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 11. doi: 10.1002/widm.1424
- Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Benetot, A., Tabik, S., Barbado, A., ... Herrera, F. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58, 82-115. doi: <https://doi.org/10.1016/j.inffus.2019.12.012>
- Bhatt, U., Xiang, A., Sharma, S., Weller, A., Taly, A., Jia, Y., ... Eckersley, P. (2020). Explainable machine learning in deployment. In *Proceedings of the 2020 conference on fairness, accountability, and transparency* (p. 648–657). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3351095.3375624
- Bove, C., Aigrain, J., Lesot, M.-J., Tijus, C., & Detryniecki, M. (2022). Contextualization and exploration of local feature importance explanations to improve understanding and satisfaction of non-expert users. In *27th international conference on intelligent user interfaces* (p. 807–819). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3490099.3511139
- Carvalho, D. V., Pereira, E. M., & Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8). doi: 10.3390/electronics8080832
- Doran, D., Schulz, S., & Besold, T. R. (2017). *What does explainable ai really mean? a new conceptualization of perspectives*. arXiv. doi: 10.48550/ARXIV.1710.00794
- Dosilovic, F. K., Bri, M., & Hlupic, N. (2018). Explainable artificial intelligence: A survey. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 0210-0215.
- hui Li, X., Cao, C. C., Shi, Y., Bai, W., Gao, H., Qiu, L., ... Chen, L. (2022). A survey of data-driven and knowledge-aware explainable ai. *IEEE Transactions on Knowledge and Data Engineering*, 34, 29-49.
- Lakkaraju, H., Kamar, E., Caruana, R., & Leskovec, J. (2019). Faithful and customizable explanations of black box models. In *Proceedings of the 2019 aaai/acm conference on ai, ethics, and society* (p. 131–138). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3306618.3314229
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon et al. (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc.
- Mohseni, S., Zarei, N., & Ragan, E. D. (2018). A survey of evaluation methods and measures for interpretable machine learning. *ArXiv, abs/1811.11839*.
- Molnar, C. (2020). *Interpretable machine learning*. Lulu.com.
- Preece, A. D., Harborne, D., Braines, D., Tomsett, R. J., & Chakraborty, S. (2018). Stakeholders in explainable ai. *ArXiv, abs/1810.00184*.
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). *Model-agnostic interpretability of machine learning*. arXiv. doi: 10.48550/ARXIV.1606.05386
- Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR, abs/1312.6034*.
- Stiglic, G., Kocbek, P., Fijacko, N., Zitnik, M., Verbert, K., & Cilar, L. (2020). Interpretability of machine learning based prediction models in healthcare. *CoRR, abs/2002.08596*.
- Tüfekci, P. (2014). Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *International Journal of Electrical Power Energy Systems*, 60, 126-140. doi: <https://doi.org/10.1016/j.ijepes.2014.02.027>
- Wood, D. (2020, 03). Combined cycle gas turbine power output prediction and data mining with optimized data matching algorithm. *SN Applied Sciences*, 2. doi: 10.1007/s42452-020-2249-7

Long Horizon Anomaly Prediction in Multivariate Time Series with Causal Autoencoders

Mulugeta Weldezigina Asres¹, Grace Cummings², Aleko Khukhunaishvili³, Pavel Parygin⁴, Seth I. Cooper⁵, David Yu⁶, Jay Dittmann⁷, and Christian W. Omlin⁸

^{1,8} *University of Agder, Norway*
mulugetawa@uia.no
christian.omlin@uia.no

² *University of Virginia, USA*
gec8mf@virginia.edu

³ *University of Rochester, USA*
Aleko.Khukhunaishvili@cern.ch

⁴ *National Research Nuclear Univ., Russia*
pavel.parygin@cern.ch

⁵ *University of Alabama, USA*
seth.cooper@cern.ch

⁶ *Brown University, USA*
david.yu@brown.edu

⁷ *Baylor University, USA*
jay.dittmann@baylor.edu

ABSTRACT

Predictive maintenance is essential for complex industrial systems to foresee anomalies before major system faults or ultimate breakdown. However, the existing efforts on Industry 4.0 predictive monitoring are directed at semi-supervised anomaly detection with limited robustness for large systems, which are often accompanied by uncleaned and unlabeled data. We address the challenge of predicting anomalies through data-driven end-to-end deep learning models using early warning symptoms on multivariate time series sensor data. We introduce AnoP, a long multi-timestep anomaly prediction system based on unsupervised attention-based causal residual networks, to raise alerts for anomaly prevention. The experimental evaluation on large data sets from detector health monitoring of the Hadron Calorimeter of the CMS Experiment at LHC CERN demonstrates the promising effi-

cacy of the proposed approach. AnoP predicted around 60% of the anomalies up to seven days ahead, and the majority of the missed anomalies are abnormalities with unpredictable noisy-like behavior. Moreover, it has discovered previously unknown anomalies in the calorimeter's sensors.

1. INTRODUCTION

Modern industrial systems utilize sensors to monitor physical quantities such as voltages, currents, flows, temperature, pressure etc. These measurements monitor system state by detecting deviations from normal operating conditions. As one of the pillars of Industry 4.0, Predictive Maintenance (PdM), which primarily depends on early anomaly detection, aims at predicting critical anomalies of a system to improve asset availability by actuating early maintenance before major system faults (Langone, Cuzzocrea, & Skantzos, 2020). Anomaly prediction is an extension of anomaly detection (AD) and focuses on predicting anomalies from early symptoms. It has cost saving potential for large complex systems through prevention of unforeseen system faults, unplanned

Mulugeta Asres et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

downtimes, and maintenance (Wagner & Hellingrath, 2021; Tang, Chen, Bao, & Li, 2019; Huang, Wu, & Wang, 2016; X. Li, Zhang, Ding, & Sun, 2020; Langone et al., 2020). However, most of the data-driven PdM models in the literature employ supervised approaches that require prior labeled anomalies and are limited to short-range predictions (Tang et al., 2019; Huang et al., 2016; X. Li et al., 2020; Langone et al., 2020; Wang, Liu, Zhu, Guo, & Hu, 2018; Hadj-Kacem, Jemaa, Allio, & Slimen, 2020).

In this study, we strive to predict anomalies through data-driven machine learning models from early warning patterns on unlabeled multivariate time series data sets. We propose AnOP, an end-to-end **Anomaly Prediction** system using unsupervised long sequence time series forecasting and anomaly detection mechanisms. The proposed system consists of a pipeline of multivariate time series autoencoder models, i.e., a long horizon sequence-to-sequence (S2S) time series forecasting (TSF) model and an AD model. The underlying concept employs a TSF model, trained on the interaction of multivariate sensor signals, to predict future temporal segments, and then uses an AD model to evaluate the predicted signals for potential anomalies. Furthermore, since additive outliers (transient and interpreted on short time scales) are generally unpredictable, our study aims at forecasting anomalous temporary changes that persevere for a certain period (multiple time steps) (T. Wen & Keyes, 2019).

As a use case study for anomaly prediction, we have discussed the Hadron Calorimeter (HCAL) of the CMS experiment at CERN. We have developed the AnOP system to predict anomalies from the multivariate diagnostics sensor data and leverage the health monitoring prognostics of the HCAL's Endcap. Capturing anomalies that persist for substantial periods, often manifested in decaying or growing trends, strange dips, or peaks, are the prime focus of the proposed system. We assessed the performance of the AnOP system in predicting temporal discords using various long sequence horizons on thirty-four Readout Boxes. Because of the lack of labeled anomaly data, we scrutinized the performance in forecasting anomalies with classification metrics as compared with the anomaly flags generated by the AD model when the true signals (non-forecasted) are supplied to it directly without the TSF model. Besides, we have incorporated an evaluation of the forecasting accuracy of the TSF model. Furthermore, we have demonstrated that the proposed system has revealed anomalies that have never been captured before in the HCAL.

The key contributions of our work are highlighted below:

- We present a data-driven unsupervised anomaly prediction mechanism, from heterogeneous multivariate time series sensor dataset.
- We introduce a time block-based S2S TSF model that captures temporal causal interactions for long sequence

multivariate time series prediction.

- We present a first study on early prognostics through data-driven methods for the HCAL Endcap Readout Box (RBX) monitoring from diagnostic sensor data.

We discuss background on anomaly prediction and the HCAL system in Section 2, and highlight the data sets used in the study in Section 3. We present the methodology of the proposed AnOP system and modeling approach in Section 4. Section 5 provides performance evaluation in long sequence forecasting and anomaly prediction on the HCAL sensor data sets. Finally, Section 6 offers our conclusion.

2. BACKGROUND

This section discusses background on anomaly prediction, multi-timestep forecasting, and the HCAL system.

2.1. Time Series Anomaly Prediction

Inadequate maintenance techniques can reduce the overall productive capacity of equipment by up to 20%, and unplanned downtimes and reactive maintenance in industrial systems incur substantial costs each year (Kamat & Sugandhi, 2020). PdM applications often refer to performing anomaly detection, diagnostics, and prognostics taking into account the Prognostics and Health Management (PHM) algorithms (Wagner & Hellingrath, 2021).

Conventionally, industries carry out PdM using statistical tests, rule-based alerts, and preset threshold limits (Rezvanizani, Dempsey, & Lee, 2014). Owing to the current advancement in sensor and data processing technologies, recent PdM approaches emphasize on machine learning approaches to capture intricate hidden patterns (Wang et al., 2018; X. Li et al., 2020; Wagner & Hellingrath, 2021). However, the existing data-driven approaches for PdM revolve around the development of supervised models which aim at specific labeled data or/and rely on feature extraction signal processing tools such as variants of Fourier transform, Wavelet transform, statistical based and principal component analysis (PCA) (Tang et al., 2019; Huang et al., 2016; X. Li et al., 2020; Langone et al., 2020; Wang et al., 2018; Hadj-Kacem et al., 2020; Hamaide & Glineur, 2021). In (Hadj-Kacem et al., 2020), a machine learning-based anomaly prediction model was proposed using forecasting future time steps mechanism for mobile networks. However, the approach covers short sequences (forecast up to a 16-step horizon) and relies on linear regression, PCA, and supervised logistic regression. Moreover, the efforts on automated feature extraction, via end-to-end deep learning, for prognosis mainly focus on remaining useful time (RUL) estimation (Gugulothu et al., 2017). Generally, the adoption of the above methods for multivariate complex systems is constrained due to high-cost data labeling on heterogeneous sensors. Besides, early signs of anomalies are often not easily seen by experts and

are challenging to annotate in large data sets from numerous monitoring sensors. Furthermore, operational quality-altering anomalies, which do not lead to an ultimate breakdown, are often overlooked. Therefore, unsupervised end-to-end deep learning methods are essential for anomaly prediction system development. Our AnOP approach employs unsupervised models and provides much longer horizon forecasting by capturing non-linear temporal interactions among multidimensional sensors via deep learning models. It determines when a system anomaly will happen, the nature of the anomaly pattern, and the affected sensors.

2.2. Long Sequence Time Series Forecasting

Many real-world applications require long sequence time series predictions, such as price forecasting in the stock market (Y. Liu, Gong, Yang, & Chen, 2020), e-commerce sell prediction (R. Wen, Torkkola, Narayanaswamy, & Madeka, 2017), traffic forecasting (Y. Li, Yu, Shahabi, & Liu, 2017), electricity consumption projecting (Y. Liu et al., 2020; R. Wen et al., 2017; Cinar et al., 2017), weather forecasting (Y. Liu et al., 2020; Cinar et al., 2017) etc. To forecast long sequence time series signals, a model with a high prediction capability (the ability to capture long-range dependencies between predictor and target data effectively) is required (Zhou et al., 2021).

Generally, long sequence forecasting approaches employ S2S autoencoder paradigm using recurrent neural network (RNN) variants (Y. Li et al., 2017; Y. Liu et al., 2020; Qin et al., 2017; Cinar et al., 2017; R. Wen et al., 2017) and Transformer (Zhou et al., 2021). However, RNN-based models may have potential limitations in inference speed and accuracy when sequence length increase due to the recursive step-by-step inferencing (Zhou et al., 2021), and in performance because of deterioration when the length of the input sequence increases (Cho et al., 2014). To address these challenges, decoder models with parallel generation are proposed using attention mechanisms (Y. Liu et al., 2020; Qin et al., 2017; Cinar et al., 2017), multilayer-perceptron (MLP) (R. Wen et al., 2017) and Transformer (Zhou et al., 2021). Nevertheless, these approaches operate only with predefined short horizons (fewer than approximately 40 data points) that limits their scalability (Z. Liu, Loo, & Pasupa, 2021; Y. Li et al., 2017; Y. Liu et al., 2020; Qin et al., 2017; R. Wen et al., 2017) except in (Zhou et al., 2021). Zhou et al. (Zhou et al., 2021) demonstrated the efficacy of an Informer model, a Transformer autoencoder architecture, with various horizons in univariate and multivariate time series data sets. However, the Informer model still lacks S2S generation for longer horizons and requires training of separate models for each target horizon. Besides, sensor data in the real world scenarios are accompanied by missing or invalid values that often results in reading segments with variable length. Hence, incorporating RNN variant models remains relevant for dealing such variability in a time series data.

2.3. Readout Boxes of the HCAL Detector

The CMS Experiment is one of the two general purpose detectors operating at CERN’s Large Hadron Collider (LHC) (Collaboration et al., 2008). The Hadron Calorimeter (HCAL) of CMS is responsible for measuring the energy of hadronic showers originating from the LHC collisions. The HCAL is divided into four subdetectors: the HCAL Barrel (HB), HCAL Endcap (HE), HCAL Outer (HO), and HCAL Forward (HF). This paper discusses only the monitoring data of the HE subdetector, a brass and scintillating plastic sampling calorimeter.

The HE is arranged into two hemispheres, HE Plus (HEP), and HE Minus (HEM). Each half is further divided into eighteen identical wedges. Signal from each wedge is read out by one “Readout Box” (RBX). Figure 1 showcases the RBX numbering and HE geometry. The RBX represents the smallest unit of front-end control and power, with each RBX consisting of a water-cooled aluminum shell housing the front-end data acquisition, control, and communication electronics. The electronics consist of low voltage distribution, high voltage distribution, four Readout Modules (RMs), a Calibration Unit (CU) and one next-generation Clock, Control, and Monitoring Module (ngCCM). The ngCCM provides backend-to-frontend communication, control, and clock distribution.

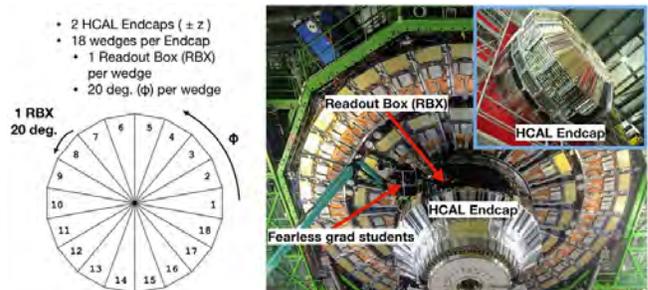


Figure 1. The HE subdetector of the CMS Experiment. Left: arrangement of eighteen RBXes. Right: installation position of the HE on the CMS detector.

To maintain physics data acquisition quality, predicting faults of the detector electronics is essential. Currently, the CMS HCAL only uses automated monitoring for general detector safety through established Detector Control and Detector Safety Systems (DCS and DSS, respectively). These systems use a small subset of the available monitored variables available to generate threshold-based alerts on quantities like temperature or bias voltage. Therefore, machine learning models have been explored for system monitoring automation of the CMS detectors through time series anomaly detection (Asres et al., 2021; Paltenghi, 2020; Azzolin et al., 2019; Wielgosz, Skoczen, & Wiatr, 2018; Wielgosz, Mertik, Skoczeń, & De Matteis, 2018). Anomalous behavior in additional variables can also indicate future detector performance issues, and escape the DCS and DSS monitoring. For

example, the gradual decrease in the monitored Received Signal Strength Indicator (RSSI) current, which is proportional to the received light at the front end from the back end optical communication links, preceded control communication loss during operation in 2018 and 2019 (Cummings & the CMS Collaboration, 2021). RSSI was not actively monitored, and trends such as depicted in Figure 2 could have been predicted. The proposed approach in this paper attempts to detect such anomalies from early signs before they affect data quality or result in loss of data.

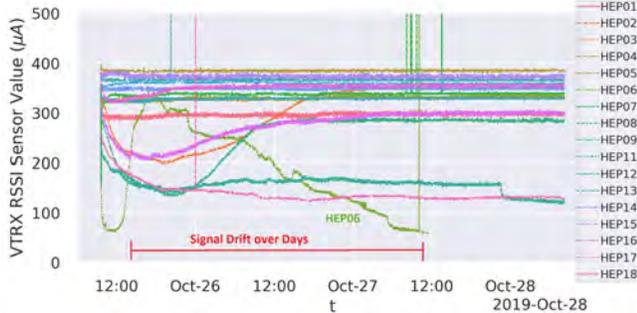


Figure 2. Gradual drifting anomalies on RSSI before ngCCM lost communication in 2019. A strong decay over three days is illustrated for the HEP06 RBX.

3. DATA SET DESCRIPTION

In this study, we have utilized front-end electronics diagnostic sensor data from the HCAL. These data sets are recorded for detector health monitoring and diagnostic purposes, not for physics data analysis. We have used ngCCM monitoring data from the HE subdetector collected in 2018 using the HE monitoring service. The HE monitoring service communicates to the front-end electronics through the ngCCM server, a software that handles access to the ngCCM. The data set contains 86M readings of around 2600 monitored quantities, measured once per minute, from 34 active RBXes (HEP01–18 and HEM01–18, excluding HEM15 and HEM16) from September to December 2018. The signals are composed of current, voltage, and optical power measurements of various components of the ngCCM. Finally, we downsampled the data into hourly intervals by averaging to capture the relevant temporal information.

4. METHODOLOGY

This section provides the methodology of the proposed anomaly prediction approach and models.

The proposed AnoP system is composed of two multivariate time series autoencoder models combined in a pipeline, i.e., i) a multi-timestep TSF model, and ii) an AD model (see Figure 3). We have discussed below the mathematical formulation and model architectures for the TSF and AD of the AnoP in Section 4.1 and Section 4.2, respectively. Section 4.3 elab-

orates the data preprocessing, preparation of training data sets and model training.

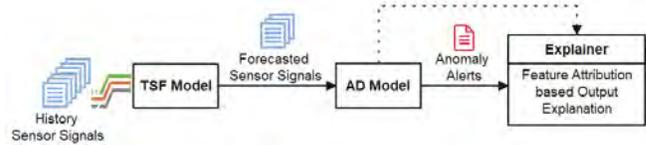


Figure 3. System design of the proposed AnoP system. The TSF model predicts a long sequence of signals, and the AD model produces anomaly status of the predicted signals based on reconstruction scores. The explainer yields explanation for the detected anomalies using post-hoc feature attribution estimation.

4.1. Multivariate Multi-timestep Forecasting Model

For long sequence forecasting, we propose a robust attention-based S2S dynamic conditional decoding mechanism. In essence, a TSF model needs to cope with two challenges in anomaly prediction. First, it should predict the deviating signals belonging to anomalies from their early fluctuation patterns. Second, it should also quickly adjust its prediction after intervention or maintenance, when normal system behavior is resumed. To achieve these capabilities, we integrate a conditional decoder for the TSF model where the latest time window of the sensors is used as conditional input. The conditional decoding enables the TSF model to respond faster when the sensor signals begin to evolve. Additionally, we employ dynamic decoding, a recursive conditional decoder, to allow dynamic long-horizon forecasting. Dynamic conditional decoding is a mechanism in which earlier slices or tokens from the model output are supplied into the decoder as conditional input to generate the subsequent output sequence. This approach has been successfully applied with S2S models in natural language processing domains such as language translation (Sutskever, Vinyals, & Le, 2014; Devlin, Chang, Lee, & Toutanova, 2018). Conditional decoding without the recursive dynamic decoding has also been extended into time series data sets in recent studies (Zhou et al., 2021).

4.1.1. Mathematical Formulation

Let the input time series data is $x^T \in \mathbb{R}^{N_x \times T}$ where N_x is the number of input sensors with a history sequence of $t_x \in [t' - T, t']$, with length of T . The TSF model F predicts the sequence $y^H \in \mathbb{R}^{N_y \times H}$ with a horizon time window of $t_y \in [t' + 1, t' + H]$ for N_y target sensors. Since, the TSF model employs S2S autoencoder, the encoder F_e maps the input x^T into context z_e and state vectors h_e Eq. (1):

$$z_e, h_e = F_e(x^T) \quad (1)$$

The decoder F_d utilizes dynamic conditional decoding that uses the context vectors z_e and conditional input sequences from the target sensors y_d from the last time steps $t_d \in [t -$

$T_d, t]$ with a size of T_d to predict the multi-timestep signals y^H and generate decoding state h_d Eq. (2):

$$y^H, h_d = F_d(y_d, z_e, h_d) \quad (2)$$

When inferencing long sequence horizon $H_l > H$ with size of l , the decoder uses dynamic decoding that behaves in an autoregressive manner employing a time block-based S2S approach (see Algorithm 1). The decoder initializes its states h_d from the encoder states, $h_d = h_e$, and then recursively predicts multi-timestep signal segments of the size H (from line 7 to 11 in Algorithm 1). The latest predicted horizon y^H is combined with the y_d to form a new conditional input to the decoder for the subsequent forecasts (line 9).

Algorithm 1 Multistep Forecasting Inference

```

1: procedure TIMEBLOCKS2SMULTISTEPFORECASTING( $F, x, y_d, H_l$ )
  ▷  $F$  : forecasting S2S encoder-decoder model
  ▷  $x$  : multivariate input times series signals with size of  $N_x \times T$ 
  ▷  $y_d$  : initial decoder input from past time-window of the target signals
  ▷  $H_l$  : time length of the target horizon
2:    $H \leftarrow \text{getModelHorizonSize}(F)$ 
3:    $N_i \leftarrow H_l/H$  ▷ number of forecasting iterations with basic block
  of  $H$ 
4:    $z_e, h_e \leftarrow F_e(x)$  ▷ get the learned context vectors and states from
  the encoder
5:    $h_d \leftarrow h_e$  ▷ initial state of decoder
6:    $y \leftarrow []$ 
7:   for  $i$  in  $[1, \dots, N_i]$  : do
8:      $y^H, h_d \leftarrow F_d(y_d, z_e, h_d)$ 
9:      $y \leftarrow \text{join}(y, y^H)$  ▷ concatenate on the time dimension
10:     $y_d \leftarrow \text{getCondInput}(y^H, y_d)$  ▷ update conditional input
11:  return  $y$ 
12:  procedure GETCONDINPUT( $y^H, y_d$ ) ▷ returns decoder
  conditional input segment
13:     $H \leftarrow \text{length}(y^H)$ 
14:     $T_d \leftarrow \text{length}(y_d)$ 
15:    if  $H \leq T_d$  then
16:       $y_d \leftarrow \text{join}(y_d\{t \in [H, T_d]\}, y^H)$  ▷ update the latest  $H$ 
  steps of  $y_d$  from  $y^H$ 
17:    else
18:       $y_d \leftarrow y^H\{t \in [H - T_d, H]\}$  ▷ get the latest  $T_d$  steps
  from the  $y^H$ 
19:    return  $y_d$ 

```

Furthermore, to improve attentiveness of the conditional inputs and leverage the multi-timestep forecasting accuracy, the decoder employs a multi-attention mechanism (see Figure 4). The model is composed of three parallel attention layers; one for the encoded latent or context vectors z_e , and two blocks for the conditional multivariate sensor signals y_d on the feature (sensor quantity) and time dimensions, respectively Eq. (3):

$$\begin{aligned} \psi_{z_e} &= \text{softmax}(z_e) \\ \psi_{y_d^t} &= \text{softmax}(y_d^t) \\ \psi_{y_d^f} &= \text{softmax}(y_d^f) \end{aligned} \quad (3)$$

where ψ_{z_e} is attention on the learned encoder context vector z_e , and $\psi_{y_d^t}$ and $\psi_{y_d^f}$ are attention scores of the decoder conditional input y_d on its temporal and feature dimensions, respectively. Finally, attention scores are concatenated to form predictor features for the multi-timestep forecasting Eq. (4):

$$\psi = [\psi_{z_e} || \psi_{y_d^t} || \psi_{y_d^f}] \quad (4)$$

4.1.2. Model Architecture

The proposed TSF S2S autoencoder model is composed of residual dilated convolutional and GRU networks with attention (see Figure 4).

To achieve temporal causation learning, multiple convolutional layers are stacked in the network with increasing dilation size. The increasing dilation along subsequent layers expands the receptive field of the convolution operation in the time data (Bai, Kolter, & Koltun, 2018; He & Zhao, 2019). Furthermore, to mitigate the performance degradation for long input sequences, we have ameliorated the model with time dimension reduction through multilayer pooling. Moreover, residual skip connections are added in the convolutional network to enhance training with deep layers.

Unlike the encoder, the decoder utilizes an attention-based network that takes decoding inputs from the encoded latent features and conditional signals. Nevertheless, the remaining sections of the decoder consists of similar blocks as the encoder but in reverse order and in deconvolution configuration. It also employs a final deconvolution layer with unit kernel size for output stabilization. Generally, the number of convolutional blocks on the encoder and decoder may differ since the encoder attempts to learn relevant context from the history time window, whereas the decoder's purpose is to predict the signals in the horizon time window. Furthermore, the conditional input signals to the decoder pass through a convolutional embedding block, to extract relevant temporal features, before the attention network. Unlike previous studies (Qin et al., 2017; Zhou et al., 2021), the attention network in our model is not followed with a fully-connected layer to reduce model complexity. It is directly connected to the GRU network, and the input weights of the first GRU layer can provide a similar functionality as fully-connected layer.

4.2. Multivariate Time Series Anomaly Detection Model

The AD model employs variational autoencoder G that attempts to reconstruct \bar{x}^T from a multivariate input data $x^T \in \mathbb{R}^{N \times T}$ from N sensors on a time sequence $t \in [t' - T, t']$. The encoder of the model provides normally distributed low-dimensional representation latent signals z Eq. (5). The decoder generates the reconstructed signals \bar{x}^T from encoded latent signals Eq. (6):

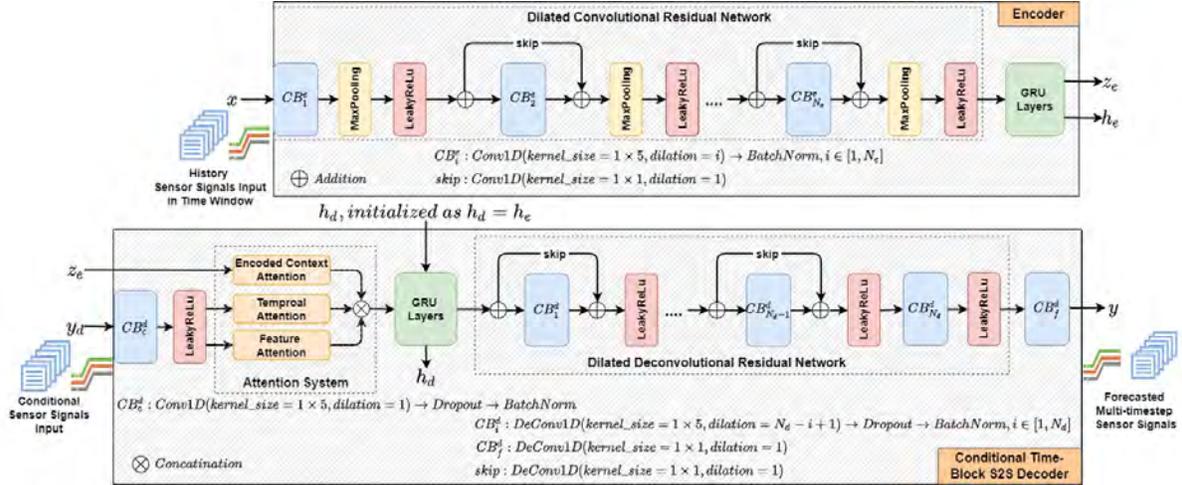


Figure 4. Architecture of the multi-timestep forecasting S2S autoencoder of the TSF model. The residual block: consists of a 1D *dilated convolutional* network while the recurrent neural network contains two *GRU* (*encoder hidden_size*: $16 \rightarrow 16$, *decoder*: $16 \rightarrow 256$) layers. The convolutional block: 1D *dilated convolutional* (256 kernels, except CB_c^d and CB_f^d with 16 and N_y kernels, respectively) for fast localized feature extraction, *BatchNorm* for network weight regularization and faster convergence, *LeakyReLU* for non-linear activation, and *MaxPooling* for prominent features retrieval that are insensitivity to time translation. *Softmax*: builds the attention in the decoder. Finally, *Dropout*=0.20 for further training regularization. Temporal causal learning via the convolutional layers with varying size of dilation and the GRU layers.

$$z = G_e(x^T) \quad (5)$$

$$\bar{x}^T = G_d(z) \quad (6)$$

Finally, the model estimates anomaly scores from the signal reconstruction errors. For each univariate sensor, reconstruction anomaly scores at time t' are calculated based on Mean Absolute Error (MAE) Eq. (7):

$$a_i(t') = \frac{1}{T} \sum_{t=t'-T}^{t'} |x_i(t) - \bar{x}_i(t)| \quad (7)$$

where x_i and \bar{x}_i are the input and reconstructed signals of the i^{th} sensor. The multidimensional reconstruction score is finally converted into system anomaly score using Mahalanobis distance (D_{md}) estimation, multidimensional distance between a point (vector) and a distribution (De Maesschalck, Jouan-Rimbaud, & Massart, 2000) Eq. (8).

$$D_{md} = \sqrt{(A_i - \mu)^T \cdot C^{-1} \cdot (A_i - \mu)} \quad (8)$$

where D_{md} is the Mahalanobis distance. The vector A_i is the multivariate anomaly score of the i^{th} observation, the vector μ contains the mean values of the univariate scores (across all observations), and C^{-1} is the inverse covariance matrix of A . Finally, a threshold $K_{md} = \alpha_{md}\mu_{md}$ is applied on the D_{md} to generate anomaly flags. The $\mu_{md} = \mathbb{E}[D_{md}]$ is the mean

distance, and α_{md} contains the adjustable parameters to tune detection sensitively.

Finally, the unsupervised autoencoder is built on 1D convolutional and GRU networks, accompanied by a post-hoc anomaly explainer based on feature attribution algorithms such as *Integrated Gradient* and *SHAP* (see Figure 5). The model is adopted from our previous work on multivariate AD for the HCAL sensor diagnostics, and further description and performance evaluation on the model can be found in (Asres et al., 2021).

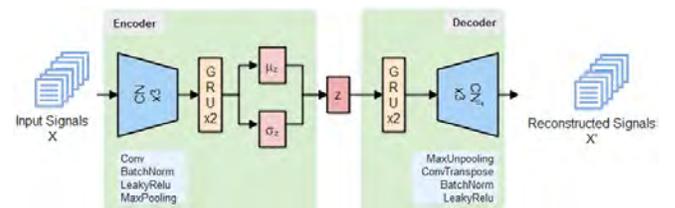


Figure 5. Architecture of the multivariate reconstruction autoencoder of the CGVAE AD model. The convolutional network is consists of three blocks; each consists of 1D *convolutional layer* (64 kernels, *kernel_size*= 1×3). The recurrent network consists of two *GRU* (*encoder hidden_size*: $16 \rightarrow 4$, *decoder*: $4 \rightarrow 16$) layers. μ_z and σ_z are fully-connected linear layers implementing the *variation layer* and $z = \mu_z + \sigma_z \odot \epsilon$, where $\epsilon \sim N(0, I)$ and \odot signify an element-wise product.

4.3. Dataset Preparation and Model Training

Since the TSF and AD models of the proposed AnOP system require different training data sets, the models were

trained separately. The AD needs a training dataset with healthy instances or low anomaly contamination, while the TSF requires substantial predictable anomalies in its training dataset. However, obtaining clean data of healthy instances in the training data is one of the main challenges of semi-supervised learning of AD models (Munir, Siddiqui, Dengel, & Ahmed, 2018). We cleaned the potential outliers from each univariate sensor data in the training set using state-of-the-art time series outlier detection algorithm, Saliency Residual (SR) (Asres et al., 2021; Zhao et al., 2020). On the other hand, the TSF autoencoder was trained on the dataset contaminated with anomalous patterns to leverage its capability to forecast anomaly signals from early signs. The modeling approach is fully unsupervised and does not require any labeling. However, since anomalies are rare instances, the model may struggle to learn the anomaly signals due to the class imbalance. We attempted to mitigate the challenge with support of the AD model. We selected the data sources, the RBXes, that have a significant number of outliers (potential anomalies) spanning substantial periods on the sensor data.

Finally, we trained the autoencoder models with *Adam* optimizer using a *super-convergence cyclic* learning rate scheduling mechanism (Smith & Topin, 2019). To mitigate Kullback-Leibler (KL) divergence vanishing or latent squashing for the variational autoencoder of the AD model, we have applied a *cyclic annealing* method (Fu et al., 2019) when KL divergence loss regularizes the training cost function.

5. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we present the performance of the proposed long sequence time series forecasting model and the AnoP system, and finally, share ideas for future research directions.

As discussed in Section 4, we trained the TSF and AD models of the AnoP on different data sets with twenty-six sensors per RBX. In our experiment, we have used the same sensors for the input and target, $N_x = N_y$. The TSF autoencoder was trained on two-month data, 10–11/2018, from six RBXes (HEM01, HEM04, HEM17, HEP14, HEP15, and HEP18), while one-month data, 10/2018, from four stable RBXes (HEM01, HEM07, HEM17, and HEP11) were used to train the AD autoencoder. The models were developed with PyTorch and trained up to 5000 iterations. Finally, we have evaluated performance of the proposed models on the date range of 25/09–03/12/2018 for thirty-four RBXes.

The TSF uses a $T = 120$ hours (5 days) sliding history time-window with prediction horizon sizes of $H = [24, 168]$ hours (1 to 7 days). The conditional decoder of the model uses the last $T_d = 24$ hours from the history time window for the target sensors. The AD model predicts anomalies on the 24 hours sliding window. We have set $\alpha_{md} = 10$, determined heuristically, to estimate the anomaly detection decision thresholds for the reconstruction anomaly detection.

Finally, we compared the anomaly prediction performance of the AnoP with the benchmark CGVAE AD model. The benchmark model is the same as the AD model of the AnoP except it detects the anomaly from the raw sensor signals in contrast to the AnoP, where the AD model detects anomalies from the forecasted signals.

5.1. Multi-timestep Forecasting Model Evaluation

In this section, we present the results on performance evaluation of the TSF model in forecasting long horizon sequences.

The model employs $N_{cd}^e = 2$ and $N_{cd}^d = 4$ casual residual convolution blocks for the encoder and decoder networks, respectively, and basic forecasting horizon $H = 24$ hours. We assessed the efficacy on multiple long horizon sizes, i.e., 24 to 168 samples (see Table 1). The results demonstrate that the model forecasted long horizons with slight performance degradation through time block S2S mechanism.

Table 1. Multivariate time series forecasting performance, averaged from all RBXes, on different horizons.

Horizon (H)	24h	48h	72h	96h	120h	144h	168h
MAE	0.418	0.430	0.444	0.464	0.473	0.503	0.529
MSE	1.392	1.416	1.465	1.515	1.558	1.635	1.705

MAE - Mean Absolute Error, MSE - Mean Square Error

Figure 6 illustrates the forecasting capability of the proposed attention mechanism with the conditional decoding as compared with conditional decoder without attention. The mean absolute error (MAE) and mean square error (MSE) performance improve substantially by 10–15% and 22–28%, respectively. Furthermore, Figure 7 portrays an ablation study on the TSF model demonstrating the promising contribution of the major building blocks of the model, i.e. the attention, conditional decoding, and convolution layers.

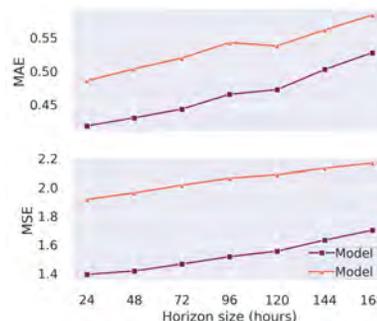
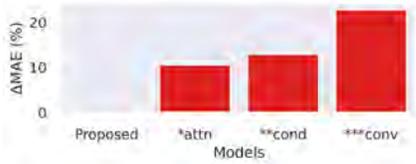


Figure 6. Multivariate time series forecasting performance comparison between different model configurations. *Model 1: the proposed attention-based conditional decoder*, and *Model 2: conditional decoder without attention*.

5.2. Anomaly Prediction Performance

In the absence of annotated data, we define an anomaly as an outlier that deviates from the expected nominal charac-



* is number of excluded blocks from the proposed TSF model

Figure 7. Ablation performance evaluation of the TSF model at $H = 24$ hours. The MAE score difference in percentage is give relative to the proposed model. *attn – w/o attention, **cond – w/o conditional decoding, and ***conv – w/o convolution layers.

teristics. Thus, not all anomalies indicate failure in the detector. The efficacy of the AD model was assessed as compared with benchmark error-counter variables of the HCAL in (Asres et al., 2021). However, the counters are less convenient to be used for anomaly prediction evaluation as they are ineffective in capturing most of the gradual system deterioration anomalies (Asres et al., 2021). Hence, we generated reference anomaly labels from the AD model, i.e., AD on the raw data (not forecasted) to assess the performance of the proposed anomaly prediction system.

Generally, on average, the AD model flagged around 160 anomalous reading points per RBX on the raw data, monitored from twenty-six sensors over a period of 10 weeks (see Figure 8). Exceptionally, higher number of flags were generated from a few RBXes due to higher variability on the readings from 1V2_CURRENT sensor on the slave control card of ngCCM (see discussion below at the end of this section).

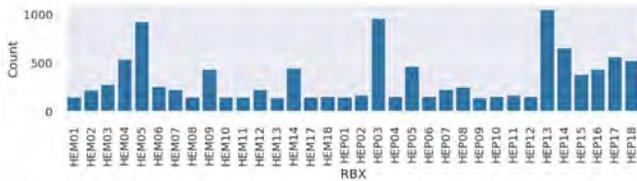
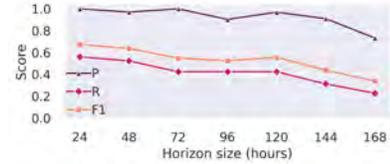


Figure 8. Number of anomaly data points, detected by the CGVAE AD model, that are used as reference flags for the evaluation of the anomaly prediction system. High number of anomalies in some RBXes such as HEM05, HEP03, HEP05, HEP13, and HEP14 due to noisy behavior of the 1V2_CURRENT sensor of the ngCCM slave control card.

Figure 9 and 10 portray the classification performance on prediction accuracy of the proposed AnoP system. The AnoP has predicted long horizon anomalies with high precision, demonstrating the robustness of the proposed system in avoiding false flags (see Figure 9). Despite this good performance, the recall is just below 0.60. This limitation is due to missed anomalies arising from unpredictable transient behavior. Additive noise is a prime cause of transient anomalies.

Our models revealed a noisy behavior of the 1V2_CURRENT sensor of the ngCCM slave control card of some RBXes. Figure 11 illustrates an example of the sensor’s behavior and our



* P - Precision, R - Recall, F1 - F1-score

Figure 9. Anomaly prediction performance of AnoP as compared to the CGVAE AD model across different horizons.

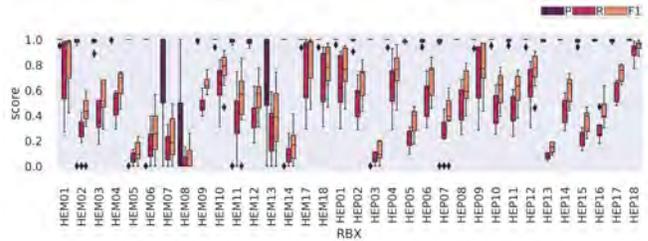


Figure 10. Distribution of anomaly prediction performance of AnoP on multiple horizons across RBXes. The lower performance in some RBXes is generally due to additive transient anomalies and noisy slave control card sensors. HEM08 has missing sensor which impacts the prediction (data was imputed with nominal value).

AnoP model’s response. The AD model generated substantial anomaly flags for those particular RBXes (see Figure 8), but the AnoP struggled to achieve good anomaly forecast (low recall) due to lack of learnable causal patterns (see Figure 10). While this was the first observation of this phenomenon in the HCAL, the behavior is not entirely unexpected. The slave control card can be noisier than the master due to the mounted FPGA’s attempt to lock onto a non-existent incoming data-stream, since the slave card does not maintain the backend communication link. This behavior does not impact operation, but monitoring its status would provide relevant information when the decision of switching the master ngCCM control card is made.

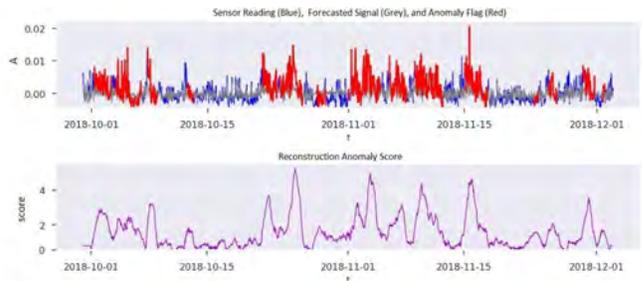


Figure 11. Anomaly prediction on 1V2_CURRENT sensor of the ngCCM slave control card of RBX HEP05. The sensor is found to be noisier in some RBXes. For instance, five times stronger noise-like fluctuation (around 0.01A) was observed as compared to its corresponding master card (0.002A) and slave card of the other RBXes. The sensor contributed a large number of anomaly flags for the RBXes. The value of the y-axis is normalized reading after subtracting the mean value across the period.

Persistent anomalies are often indicators of severe problems in the monitored system, and Figure 12 portrays a captivating anomaly captured from the successful forecast of persistent outliers, i.e., in the current and voltage sensors of the RBXes from October 28 to November 03, 2018. We found that during that time there had been Machine Development (MD) and Technical Stop (TS) tasks on the LHC. The MD weeks are planned in the LHC operation schedule to optimize and study the performance of the machine and to allow the operators to improve the long-term performance of the LHC. Following our finding, investigations revealed that the MD and TS task had unexpectedly affected the low-voltage supply of the RBX. The changes were within tolerance, and did not negatively impact HCAL’s performance, but this knowledge allows the HCAL team to better prepare for LHC interventions in the future.

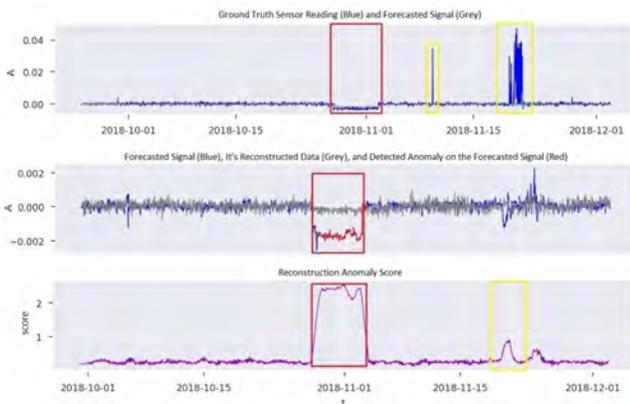


Figure 12. Forecasting capability on persistent and transient anomalies on the 1V2_CURRENT sensor of the master control card of the HEP03. (Top) forecasted signal from the TSF autoencoder using 24 hours horizon as compared to the ground truth signal. (Middle) signal reconstruction via the AD autoencoder from the forecasted signal. (Bottom) the estimation of the reconstruction-based anomaly score on the forecasted signal. Red boxes highlight the persistent outliers (successfully forecasted), whereas the yellow boxes enclose the transient or spike outliers (challenging to forecast).

5.3. Directions for Future Research

The robustness of AnoP relies on the accuracy of the employed TSF and AD models. In general applicability to sensor data with limited anomaly samples, two suggestions can be rendered generally to mitigate the class-imbalance during training of the TSF model, i.e., (i) weighted training loss functions, and (ii) data augmentation through synthetic data generation. Having an AD model beforehand, the data sets can be annotated with ease and higher weights can be assigned to the sections with anomalous patterns during training loss estimation. The other alternative is to generate and incorporate synthetic data into the training dataset (Ducoffe, Haloui, & Gupta, 2019). The recent progress on deep generative adversarial network (GAN) models has demonstrated

good capability on multivariate time series signals (Ducoffe et al., 2019; Yoon, Jarrett, & Van der Schaar, 2019).

6. CONCLUSION

Predictive Maintenance, owing to its versatile leverages in significantly cutting maintenance costs and downtimes, has become a pillar application of Industry 4.0. In this study, we have demonstrated the efficacy of the anomaly prediction approach (AnoP) through unsupervised end-to-end long time series forecasting and anomaly detection mechanisms on multivariate time series data. The experimental evaluation on the CMS HCAL diagnostic monitoring sensor data sets has unveiled that anomalies that persevere for a certain period can be forecasted from early indications. The developed anomaly prediction system is expected to enable prognostics and predictive maintenance in the HCAL during LHC RUN III. Currently, the AnoP is under pre-production testing phase for the HCAL monitoring. Finally, the proposed approaches of the AnoP are generic enough to be applied with less effort for predictive maintenance applications in other domains with time series data.

REFERENCES

Asres, M. W., Cummings, G., Parygin, P., Khukhunaishvili, A., Toms, M., Campbell, A., ... Omlin, C. W. (2021). Unsupervised deep variational model for multivariate sensor anomaly detection. In *Ieee pic* (pp. 364–371).

Azzolin, V., Andrews, M., Cerminara, G., Dev, N., Jessop, C., Marinelli, N., ... Vlimant, J.-R. (2019). Improving data quality monitoring via a partnership of technologies and resources between the cms experiment at cern and industry. In *Epj web of conference* (Vol. 214, p. 01007).

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv:1406.1078*.

Cinar, Y. G., Mirisae, H., Goswami, P., Gaussier, E., Aït-Bachir, A., & Strijov, V. (2017). Position-based content attention for time series forecasting with sequence-to-sequence rnns. In *Iconip* (pp. 533–544).

Collaboration, C., Chatrchyan, S., Hmayakyan, G., Khachatryan, V., Sirunyan, A., Adam, W., ... others (2008). The cms experiment at the cern lhc. *JInst*, 3, S08004.

Cummings, G., & the CMS Collaboration. (2021). *Cms hcals vtx-induced communication loss and mitigation*. private communications.

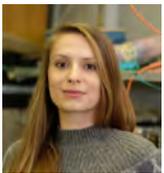
De Maesschalck, R., Jouan-Rimbaud, D., & Massart, D. L. (2000). The mahalanobis distance. *Chemometr Intell.*

- Lab. Syst.*, 50(1), 1–18.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.
- Ducoffe, M., Haloui, I., & Gupta, J. S. (2019). Anomaly detection on time series with wasserstein gan applied to phm. *IJPHM*, 10(4).
- Fu, H., Li, C., Liu, X., Gao, J., Celikyilmaz, A., & Carin, L. (2019). Cyclical annealing schedule: A simple approach to mitigating kl vanishing. *arXiv:1903.10145*.
- Gugulothu, N., Tv, V., Malhotra, P., Vig, L., Agarwal, P., & Shroff, G. (2017). Predicting remaining useful life using time series embeddings based on recurrent neural networks. *arXiv:1709.01073*.
- Hadj-Kacem, I., Jemaa, S. B., Allio, S., & Slimen, Y. B. (2020). Anomaly prediction in mobile networks: A data driven approach for machine learning algorithm selection. In *Ieeefip noms* (pp. 1–7).
- Hamaide, V., & Glineur, F. (2021). Unsupervised minimum redundancy maximum relevance feature selection for predictive maintenance: Application to a rotating machine. *IJPHM*, 12(2).
- He, Y., & Zhao, J. (2019). Temporal convolutional networks for anomaly detection in time series. In *Journal of physics: Conference series* (Vol. 1213, p. 042050).
- Huang, C., Wu, X., & Wang, D. (2016). Crowdsourcing-based urban anomaly prediction system for smart cities. In *Proceedings of acm cikm* (pp. 1969–1972).
- Kamat, P., & Sugandhi, R. (2020). Anomaly detection for predictive maintenance in industry 4.0-a survey. In *E3s web of conferences* (Vol. 170, p. 02007).
- Langone, R., Cuzzocrea, A., & Skantzos, N. (2020). Interpretable anomaly prediction: Predicting anomalous behavior in industry 4.0 settings via regularized logistic regression tools. *DKE*, 130, 101850.
- Li, X., Zhang, W., Ding, Q., & Sun, J.-Q. (2020). Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation. *Journal of Intelligent Manufacturing*, 31(2), 433–452.
- Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv:1707.01926*.
- Liu, Y., Gong, C., Yang, L., & Chen, Y. (2020). Dstp-rnn: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction. *Expert Systems with App.*, 143, 113082.
- Liu, Z., Loo, C. K., & Pasupa, K. (2021). A novel error-output recurrent two-layer extreme learning machine for multi-step time series prediction. *Sustainable Cities and Society*, 66, 102613.
- Munir, M., Siddiqui, S. A., Dengel, A., & Ahmed, S. (2018). Deepant: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access*, 7, 1991–2005.
- Paltenghi, M. (2020). *Time series anomaly detection for cern large-scale computing infrastructure* (Unpublished doctoral dissertation). Politecnico di Milano (IT).
- Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., & Cottrell, G. (2017). A dual-stage attention-based recurrent neural network for time series prediction. *arXiv:1704.02971*.
- Rezvanizani, S. M., Dempsey, J., & Lee, J. (2014). An effective predictive maintenance approach based on historical maintenance data using a probabilistic risk assessment: Phm14 data challenge. *IJPHM*, 5(2).
- Smith, L. N., & Topin, N. (2019). Super-convergence: very fast training of neural networks using large learning rates. In T. Pham (Ed.), (Vol. 11006, pp. 369 – 386). SPIE. doi: 10.1117/12.2520589
- Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Neurips* (pp. 3104–3112).
- Tang, Z., Chen, Z., Bao, Y., & Li, H. (2019). Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring. *Struct. Cont. and Health Mon.*, 26(1), e2296.
- Wagner, C., & Hellingrath, B. (2021). Supporting the implementation of predictive maintenance: a process reference model. *IJPHM*, 12(1).
- Wang, J., Liu, C., Zhu, M., Guo, P., & Hu, Y. (2018). Sensor data based system-level anomaly prediction for smart manufacturing. In *Ieee bigdata* (pp. 158–165).
- Wen, R., Torkkola, K., Narayanaswamy, B., & Madeka, D. (2017). A multi-horizon quantile recurrent forecaster. *arXiv:1711.11053*.
- Wen, T., & Keyes, R. (2019). Time series anomaly detection using convolutional neural networks and transfer learning. *arXiv:1905.13628*.
- Wielgosz, M., Mertik, M., Skoczeń, A., & De Matteis, E. (2018). The model of an anomaly detector for hilumi lhc magnets based on recurrent neural networks and adaptive quantization. *IFAC EAAI*, 74, 166–185.
- Wielgosz, M., Skoczen, A., & Wiatr, K. (2018). Looking for a correct solution of anomaly detection in the lhc machine protection system. In *Icses* (pp. 257–262).
- Yoon, J., Jarrett, D., & Van der Schaar, M. (2019). Time-series generative adversarial networks. *NeurIPS*, 32.
- Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., ... Zhang, Q. (2020). Multivariate time-series anomaly detection via graph attention network. *arXiv:2009.02040*.
- Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of aaai*.

BIOGRAPHIES



Mulugeta Weldezigina Asres is a Ph.D. candidate in Artificial Intelligence at the University of Agder, Norway. His current research is on AI models for the Hadron Calorimeter monitoring and diagnostics at the CMS Experiment at CERN. He received his B.Sc. and M.Sc. in Computer Engineering and Gold Medal Award for the highest CGPA from EiT-M, Mekelle University. He conducted post-graduate research on machine learning for non-intrusive energy monitoring, power substation systems, and telecom networks at the Polytechnic University of Turin. His research interests focus on system monitoring, time series modeling, deep learning, and Industry 4.0.



Grace Cummings is a Ph.D. candidate in Experimental High Energy Particle Physics at the University of Virginia. She received her Bachelor of Science in Physics at Virginia Commonwealth University. Her current research interest is calorimetry for particle physics with an emphasis on detector development, systems testing, and front-end instrumentation.



Aleko Khukhunaishvili is a research associate at the University of Rochester. He received his Ph.D. in Physics at Cornell University in 2014. His current research interests focus on precision Standard Model measurements at the LHC.



Pavel Parygin is a Ph.D. candidate in experimental high energy physics at the National Research Nuclear University MEPhI. He received his honors diploma in semiconductor electronics and physics of semiconductors at the National University of Science and Technology MISiS. He is a member of the CMS collaboration at the CERN and currently leading the operations group of the Hadron Calorimeter of the CMS experiment.



Seth I. Cooper is a research scientist at the University of Alabama, where he also did postdoctoral research. He received his B.A. from Carleton College in Physics and Computer Science and his Ph.D. from the University of Minnesota. Based at CERN, his current research focuses on data acquisition and online monitoring of detector systems, in addition to searches for physics beyond the standard model. He received the CMS Achievement Award in 2014.



David Yu completed his Ph.D. in Physics at the University of California, Berkeley in 2015. He is currently a senior research associate at Brown University and a distinguished researcher at the Large Hadron Collider Physics Center of the Fermi National Accelerator Laboratory. He conducts experimental high energy physics research at the CMS experiment at the Large Hadron Collider at CERN.



Jay Dittmann received his Ph.D. in Physics from Duke University, North Carolina, USA in 1998. He is currently a professor of Physics at Baylor University, engaged in experimental high energy physics research using data collected by the CMS experiment at the Large Hadron Collider at CERN in Geneva, Switzerland. He is a member of the American Physical Society.



Christian W. Omlin has been a professor of Artificial Intelligence at the University of Agder since 2018. He has previously taught at the University of South Africa, University of the Witwatersrand, Middle East Technical University, University of the South Pacific, University of the Western Cape, and Stellenbosch University. His expertise is in deep learning with a focus on applications ranging from safety to security, industrial monitoring, renewable energy, banking, sign language translation, healthcare, bioconservation, and astronomy. He is particularly interested in the balance between the desire for autonomy using AI technologies and the necessity for accountability through AI imperatives such as explainability, privacy, security, ethics, and artificial morality for society's ultimate trust in and acceptance of AI. He received his Ph.D. from Rensselaer Polytechnic Institute and his MEng from the Swiss Federal Institute of Technology, Zurich, in 1995 and 1987, respectively.

Experimental Validation of Multi-fidelity Models for Prognostics of Electromechanical Actuators

Leonardo Baldo¹, Pier Carlo Berri¹, Matteo D. L. Dalla Vedova¹, and Paolo Maggiore¹

¹ *Department of Mechanical and Aerospace Engineering (DIMEAS), Politecnico di Torino, Turin, 10129, Italy*

leonardo.baldo@polito.it

pier.berri@polito.it

matteo.dallavedova@polito.it

paolo.maggiore@polito.it

ABSTRACT

The growing adoption of electrical energy as a secondary form of onboard power leads to an increase of electromechanical actuators (EMAs) use in aerospace applications. Therefore, innovative prognostic and diagnostic methodologies are becoming a fundamental tool to early identify faults propagation, prevent performance degradation, and ensure an acceptable level of safety and reliability of the system. Furthermore, prognostics entails further advantages, including a better ability to plan the maintenance of the various equipment, manage the warehouse and maintenance personnel, and a reduction in system management costs.

Frequently, such approaches require the development of typologies of numerical models capable of simulating the performance of the EMA with different levels of fidelity: monitoring models, suitably simplified to combine speed and accuracy with reduced computational costs, and high fidelity models (and high computational intensity), to generate databases, develop predictive algorithms and train machine learning surrogates. Because of this, the authors developed a high-fidelity multi-domain numerical model (HF) capable of accounting for a variety of physical phenomena and gradual failures in the EMA, as well as a low-fidelity counterpart (LF). This simplified model is derived by the HF and intended for monitoring applications. While maintaining a low computing cost, LF is fault sensitive and can simulate the system position, speed, and equivalent phase currents.

These models have been validated using a dedicated EMA test bench, designed and implemented by authors. The HF model can simulate the operation of the actuator in nominal conditions as well as in the presence of incipient mechanical faults, such as a variation in friction and an increase of backlash in the reduction gearbox.

Leonardo Baldo et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Comparing the preliminary results highlights satisfactory consistency between the experimental test bench and the two numerical models proposed by the authors.

1. INTRODUCTION

The adoption of the so-called "more electric" paradigm is slowly but steadily reshaping aircraft design and their operations, leading to the development of new technologies. At the same time, the buildup of complex and innovative solutions requires appropriate theoretical and modelling studies to gain the relevant knowledge base and scientific know-how concerning, among other aspects, their safety.

Moreover, More Electric Aircrafts (MEAs) embrace a completely new subsystems architecture (AbdElhafez & Forsyth, 2009), which gives birth to a brand-new set of requirements, aimed at reducing weight and fuel consumption by limiting the hydraulic and pneumatic subsystem influence on the overall aircraft.

A case in point is the development and adoption of Electromechanical Actuators (EMAs), whose deployment, which is in line with the new requirements linked to MEAs, requires a solid modelling back up.

In this sense, a general overview of EMAs solutions can be found in (Qiao et al., 2018) along with the related opportunities and challenges. EMAs are emerging as lighter and more efficient solutions (Garcia Garriga, Ponnusamy, & Mainini, 2018) than hydraulic actuators for flight controls. EMA use in aircraft is hence becoming more widespread but their extensive usage is still slowed down due to the limited experience in terms of safety and reliability, being the latter extremely important especially when EMAs are used as safety critical devices. Some problems are related to their critical modes, which are usually active (e.g. mechanical jamming of the transmission) and to EMC issues, whose prediction is challenging (Balaban et al., 2009) supplies a thorough review on typical critical EMA failure modes).

In other words, they lack the knowledge base which other actuator types have (e.g. hydraulic or hydrostatic ones), in relation to fault detection, identification (FDI), prognosis, reliability and safety in general. To support EMA usage, reliable prognostic tools are required to provide a precise estimation of the system actual state and to assess the Remaining Useful Life (RUL) of components and subsystems.

Therefore, a real time monitoring system based on intricate and highly tuned algorithms is crucial to guarantee and satisfy the expected safety standards (e.g. (Berri, Dalla Vedova, & Mainini, 2021)). On the other hand, Prognostics and Health Management (PHM) systems must rely on a detailed set of numerical models, which must be correctly tuned to reproduce the actuators' behaviour in terms of static and dynamic response (e.g. currents, voltages, speed, position etc).

This is the reason why a solid and detailed modelling study comes into place and becomes pivotal to allow a more capillary use of EMA in the aerospace sector ((Berri, Dalla Vedova, & Mainini, 2022)).

Thanks to the comparison between the real response of the operating components or systems (through a precise monitoring) and the nominal response, provided by a reference model, PHM methods can predict progressive failure evolution, thus estimating components RUL. In fact, according to acquired information, various decision making strategies (e.g. Sense-Infer-Plan-Act suggested in by (Mainini & Willcox, 2015)) can be used before the hidden failure could change into catastrophic or hazardous failure conditions. De facto, RUL estimation can be exploited to enhance mission readiness: aircraft operations can be rearranged dynamically, maintenance actions can be scheduled in more convenient ways and the entire integrated logistic support architecture can be improved (Sutharssan, Stoyanov, Bailey, & Yin, 2015). Finally, extensive usage of PHM methods on board could also result in a reduction of Life Cycle Costs (LCC) due to the high cut to maintenance costs (Williams, 2006).

It is now clear that models development and the relative tuning is a crucial step for a progressive and leading-edge design. However, models validation is just as important as their creation. The outputs, trends, values and the model predictions in general have to be thoroughly validated thanks to detailed and broad experimental data sets. Should not model predictions be validated, the results can not be considered credible and reliable at any point. Test benches, like the ones available in our laboratories, are ideal platforms to obtain and collect experience, know-how and data-sets which can then be used to experimentally validate pre-build models.

To all intents and purposes, models experimental validation is a largely widespread technique harnessed in all those engineering applications which involve the development of new simulation, monitoring and control models (e.g. (Di Rito, Denti, & Galatolo, 2008) and (Bertolino, De Martin, Jacazio, & Sorli, 2020)).

In particular, in this work, the authors have focused on the modelling and validation of backlash phenomena. They covered in depth the implementation of backlash simulation in the model as well as the related experimental set-up and tests. Backlash effects are very important in a mechanical transmission as neglecting them may cause the underestimation of critical aspects, for instance the overall stability (with limit cycles) and accuracy of the system (Maré, 2017). EMA are slowly starting to earn important roles in aircrafts flight control systems (e.g. Boeing B787) and their increasing affirmation is a direct consequence several fast-growing fields of research, such as the ones involving test benches construction, PHM strategies, model development and their validation.

2. TEST BENCH DESCRIPTION

During the last years, a highly modular, compact and versatile test bench has been developed to validate numerical models and to support research activities (Figure 1). To all intents and purposes, it was built around a pre-built model, explained later on in the paper and more in details in (Berri, Dalla Vedova, & Maggiore, 2019), to provide an experimental platform which could have been able to validate it.

Some in depth layout configuration and design principles concerning the test bench are explained in details in (Berri, Dalla Vedova, & Maggiore, 2021) and (Berri, 2021). However a brief description is mandatory for reasons of clarity.

The test bench can be roughly split up into:

- Actuation Module (Light-blue block)
- Transmission Module (Green block)
- Friction Load Simulation Module (Orange block)

Despite the industrial origin of many parts, their working principles are the same of aerospace components, thus the results are still valid. Conceivable differences are taken into account in the model when possible (e.g. power density). In other cases, they are not relevant for the dynamical behaviour (e.g. redundancies) and, as such, not modelled at all. The components expected behaviour in performed test is therefore assumed to be faithful enough with respect to aerospace hardware for the validation of the models (Berri, 2021) (Giangrande et al., 2018).

2.1. Actuation Module

The actuation module is made up of an integrated Siemens motor environment, the *SI20 AC/AC Trainer Package*. This package provides full authority for the shaft motion thanks to proprietary software, inverters and control units. Through a PC (1) and the Control Unit (2), the required command is generated and then sent to the motor (3). The motor output shaft is linked to a gearbox (4) input shaft through an elastic coupling. The test bench architecture emulates an actual typical EMA implementation, with a high gear-ratio mechanical gearbox placed at the motor output shaft.

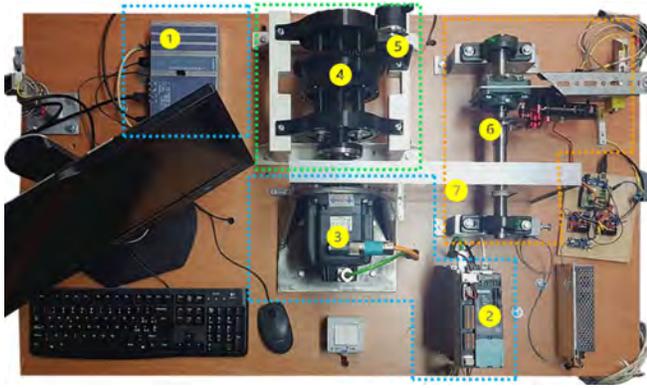


Figure 1. The test bench.

The implemented motor is a Siemens three phase, Permanent Magnet Synchronous Motor (PMSM): the S 1FK7060-2AC71-1CA0. This motor is controlled by a Siemens inverter which also logs motor electrical parameters with a frequency of $400Hz$. Table 1 shows the motor main engineering characteristics (where the notes $60K$ and $100K$ refer to an over-temperature of $60K$ and 100 respectively).

2.2. Transmission Module: the Planetary Gearbox

As stated before, the test bench involves a planetary gearbox to be consistent with real life EMAs applications.

In fact, in EMAs classic configurations, a gearbox is an essential assembly component ((Qiao et al., 2018), (De Martin, Jacazio, & Vachtsevanos, 2017), (Maré, 2017)): usually, the user requires high torques and relatively low angular speeds. On the other hand, high torque motor are generally cumbersome and heavy, consequently they are not a viable solution where lightness and compactness is an essential requirement. The solution is exploiting a smaller motor with lower available torque but higher maximum angular speed and then, through a gearbox, reducing angular speeds while increasing torque. It goes without saying that, being gearboxes essential components, they must be considered inside the diagnostic and prognostic model loops.

The high transmission ratio ($1 : 124$) gearbox has been en-

tirely designed, built and assembled inside our laboratories starting from a Wolfrom drive layout (Garcia et al., 2019). Through the careful tuning of Wolfrom drive design parameters, a lightweight, high efficiency, additive manufacturing based planetary gearbox had been obtained and explained in (Berri, Dalla Vedova, Maggiore, & Riva, June 2020). Figure 2 shows the gearbox rendering.

The motion is picked up by an external incremental encoder (5) (5000 pulses per revolution), which is mounted on a carefully designed encoder support, engaging with the external output ring of the gearbox thanks to a spur gear.

The encoder support provides variable mechanical play between the encoder and the external output ring of the transmission, thanks to a micrometer. In this way, backlash effect can be reproduced by varying the mechanical play and useful data can be saved. Hence, backlash is introduced in the test bench downstream of the gearbox, between the user output gear and the encoder.

The encoder support assembly is composed of two FDM built pieces and a micrometer as seen in the rendering in figure 3 (Baldo, 2021).

2.3. Friction Load simulation module

A friction module (Figure 4) is essential to simulate realistic conditions, for instance the presence of progressive failures or variable friction due to components wear-out.

This is achieved through a braking torque obtained with a disk brake system, controlled in closed loop with a force sensor. In fact, another steel shaft (6) is placed parallel to the gearbox and fixed to the test bench structure with two self-aligning bearing assemblies. A chain (7) (enclosed in a safety case) links the motor output shaft with the braking shaft through a sprocket and the steel disk is fixed to the braking shaft.

The brake is actuated via a simple servomotor, similar to the ones used in the model making sector. It has to be noted that, the friction block acts directly on the motor output shaft and not on the overall gearbox output. This is an arbitrary and conscious choice in order not to excessively stress the planetary gearbox (PLA based), hence strictly linked to the test bench structural characteristics.

Table 1. Test bench motor main characteristics.

Characteristic	Value
Rated speed (100 K)	2000 <i>rpm</i>
Number of poles	8
Rated torque (100 K)	5.3 <i>Nm</i>
Rated current	3.0 <i>A</i>
Static torque (60 K)	5.00 <i>Nm</i>
Static torque (100 K)	6.0 <i>Nm</i>
Stall current (60 K)	2.55 <i>A</i>
Stall current (100 K)	3.15 <i>A</i>
Moment of inertia	7.700 <i>kgcm²</i>
Efficiency	90.00



Figure 2. Gearbox rendering.



Figure 3. Encoder support assembly rendering.

The brake can generate around $20 - 30 Nm$ of torque; on the other hand the stall torque of the Siemens motor is around $6 Nm$. This would have led to the servomotor working well outside its optimal operating range with increased internal frictions, random errors, lower test repeatability etc. Thanks to the chain transmission ratio, the braking torque felt by the driving shaft (i.e. motor shaft) is lower, therefore the servomotor can work in the best conditions possible.

The servomotor output shaft is connected with a steel rod to the brake assembly, consisting of two pads which can make contact with the disk, generating friction, hence transferring an external load to the motor thanks to the steel chain. A load cell is mounted to the metal plate to measure the mechanical deformation during the tests.

The control is handled via a PI controller, which stands for "Proportional-Integral". This special category of controllers, widely used in industrial applications, employs a simple proportional logic with a constant gain and an integral action aimed at reducing the steady state error. A derivative controller is not used since in this application the error reduction at steady state is much more important than response.

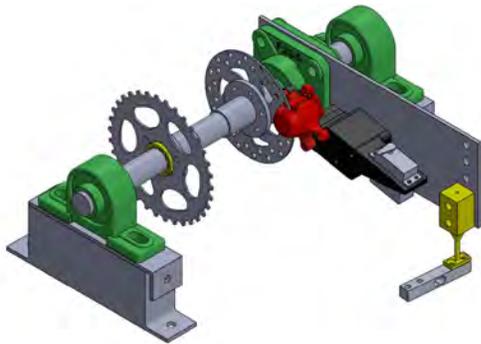


Figure 4. Braking module.

3. THE MODEL

3.1. Model Description

For the sake of clarity, a brief description of the implemented EMA model is here reported, even though an in depth explanation can be found in (Berri, 2021). Moreover, a detailed description of the proposed models as well as the the explanation of the proper purposes, objectives and limitation is reported in (Berri, 2021), (Berri et al., 2019) and (Berri et al., 2018).

The EMA model block diagram is presented in Figure 5 and it shows the main interactions between the Simulink sub models, as well as the model's inputs and outputs.

It has to be recalled that this model is further integrated into an higher level Simulink model, which takes into account the signal acquisition and filtering process, the command module, the load signal etc.

The model has been realised thanks to a physical-based approach: actual equations, which describe system physical phenomena, are implemented in Simulink blocks (e.g. EM motor equations, dynamic equations etc). Required data for the models have been taken from components data sheets or obtained thanks to experimental tests. More details on the underlying equations and data can be found in (Berri, 2021), (Sciandra, 2020) and (Boschetti, 2020).

The *set* signal is inherited from the command module, from which it is possible to generate different command shapes (e.g. ramp, step, chirp, sinusoidal) and select their main parameters. The *Flight Condition* signal is provided from a "Repeating sequence" block that provides the model with the load value at the required time (linear interpolation is used if load cell values do not have the same sample time as the HF model). The raw data are supplied from the external load cell to simulate the external torque, with a Matlab script and an Arduino Board as a low level electronic interface.

3.1.1. Actuator Control Electronics (ACE)

The Actuator Control Electronics model takes as inputs the *set* signal (position or speed) as well as the motor measured speed and user position and it is responsible for the generation of the requested torque and current.

In other words, it carries out the computation of the control law according to the selected command. This block enforces a PID controller with a low pass derivative filter, current saturation and anti-windup protection. The choice of implementing a PID controller is backed up by the enormous number of industrial applications which are based on this control strategy: this kind of controller is still the industry standard and that is the reason why it is implemented in this model. A white band noise signal is added to simulate electromagnetic noise on the command line. The output of the module is the stator current, obtained by dividing the torque (inherited from the PID controller) by the back-EMF coefficient of the motor.

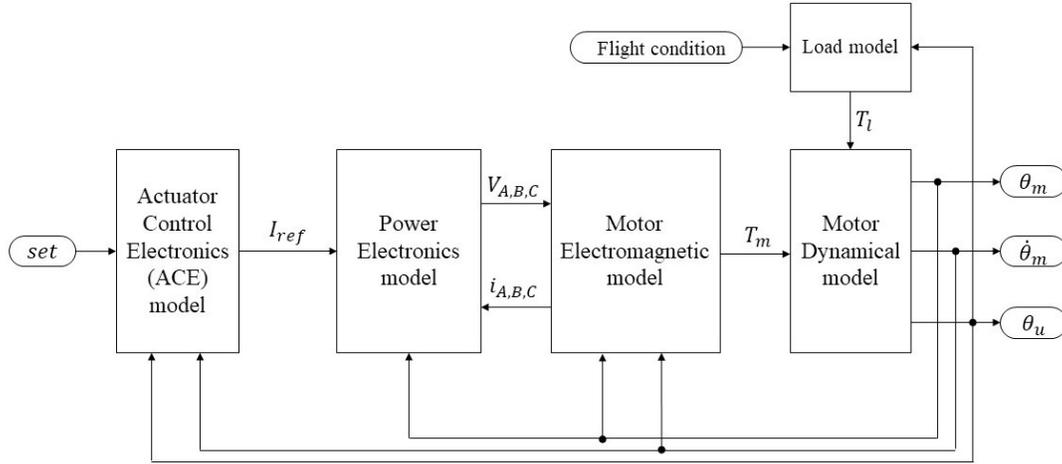


Figure 5. EMA functional block diagram as taken from (Berri,2021).

3.1.2. Power Electronics Model (PEM)

The PEM is made up of three main subsystems, each one with its separated functions:

- The first one is responsible for the evolution of the phase currents (i.e. the commutation sequence of the phases). Inverted Clarke-Park inverse transformation (with multiple reference systems) are exploited. I_{ref} is split into the three phase currents thanks to the information given from the angular position of the motor; hence the three motor phases are switched on or off, thanks to the commutation logic.
- The second module carries out the hysteresis control, comparing the stator phase currents with the respective reference currents. These currents then control the activation of the three phases' MOSFETs. If the difference between a phase current and the reference current lies outside a hysteresis band, the corresponding phase will be powered; otherwise, it will be switched off. As a matter of fact, this block generates a high frequency square wave which can be intended as a PWM signal.
- The last block consists of a Three-phase bridge, implemented via Simscape and made up of six MOSFETs with protection diodes. This block takes as input three Boolean values generated from the previous module. Extreme caution was placed in order not to create short circuit situations.

3.1.3. Motor Electromagnetic Model

The Electromagnetic model is responsible for the computation of the Back-EMF coefficient, of the currents flowing in the three phases and the calculation of the motor torque. The currents are dynamically estimated thanks to three star connected RL branches (with floating neutral) which model the

three stator phases (in Simscape). Knowing both the currents and the back-EMF coefficient at each integration step and with the assumption of linear superposition of each phase contribution, the total motor torque is easily obtainable (Eq. 1):

$$T_m = \sum_{j=A,B,C} i_j \cdot k_j \quad (1)$$

with i_j the current and k_j the back-EMF coefficient on each one of the three phases A, B, C . Finally, a saturation is placed to model magnetic flux non linearity. The total thrust is then sent to the Dynamical Model.

3.1.4. Load Model

The load module is just a simple gain which takes into account the geometry of the test bench, thus computing a torque, starting from a force measurement. De facto, the gain is the distance (arm) between the force cell and the breaking shaft explained in Section 2.3.

3.1.5. Motor Dynamical Model: Backlash

The motor and transmission dynamical model serves as the core of the simulation; it takes as inputs the motor torque and the external torque and determines the motor and users positions.

It features a second order dynamical system (Eq. 2) and, through its multiple integration, the user angular position can be obtained.

$$T_m - T_l = J_m \frac{d^2\theta_m}{dt^2} + C_m \frac{d\theta_m}{dt} \quad (2)$$

where T_m is the motor torque, T_l is the external torque (Load), J_m is the assembly inertia, C_m is the viscous friction coef-

ficient of the assembly and θ_m is the motor position. The dynamical model takes into account different non linearity:

- the effects of dry friction (implemented with the Borello model (Borello & Dalla Vedova, July 2006))
- the effects of viscous friction (with the viscous friction coefficient C_m dependent on speed)
- the effects of the assembly inertia (estimated with CAD tool)
- the effects of end-stops (detected with a saturated position integrator)
- backlash phenomena

Backlash phenomena are particularly important and, as such, they require a more in depth explanation.

Backlash is a mechanical non linearity which greatly impacts the goodness of speed and position control performance, in particular if high precision is required. It is caused by the mechanical play between parts, typically gears and causes irregularities in the transmission, since there are some moments where the gears are not touching each other and then they suddenly come into contact. Moreover, when the gap caused by the mechanical play is wide open, the output gear can not be controlled (Nordin & Gutman, 2002).

Even if usually electromechanical actuator parts are designed with a small interference, wear and degradation due to use may result in mechanical plays after some time.

Therefore, a detailed modelling of a mechanical actuator can not escape considering backlash into the control loop since excluding it may cause the underestimation of critical aspects, for instance the overall stability (with limit cycles) and accuracy of the system. As stated in (Maré, 2017), gear backlash is detrimental for the service life of the contacts and for control stability, especially when the actuators have to work in position control loops (as in primary and secondary flight control actuators).

Moreover, other negative effects of joint backlash can be detected in a mechanical transmission such as a deleterious impact on frequency response or different hazardous conditions (e.g. load oscillations leading to flight controls flutter phenomena or steerable landing gear shimmy) (Maré, 2017).

For the purpose of this model a very rough backlash modelling has been implemented thanks to a simulink block which (hysteresis band) placed on the user shaft position. The hysteresis band acts as reported in Eq. 3 (taken from (Berri, 2021)), with a band-width of $2BLK$ from the motor position.

$$\theta_u(t) = \begin{cases} \frac{\theta_m(t)}{i} + BLK, & \text{if } \theta_u(t - dt) - \frac{\theta_m(t)}{i} \geq BLK \\ \frac{\theta_m(t)}{i} - BLK, & \text{if } \theta_u(t - dt) - \frac{\theta_m(t)}{i} \leq -BLK \\ \theta_u(t - dt), & \text{otherwise} \end{cases} \quad (3)$$

Where θ_u is the user position t is the simulation time, dt is the step time, i is the gear ratio and BLK is the backlash amplitude. The "if" condition is determined by the difference between the user position at the simulation step (involving i) and the user position at the simulation step before.

This simple backlash modelling solution presents some limitations. In fact, the model outputs acceptable predictions only if the primary backlash source is deemed to be between the output gear and the encoder gear. The reason is that, only in this condition, the inertia and load downstream of the mechanical play are negligible.

In the future, a more detailed and precise backlash modelling may be taken into account: for instance, a multi-body simulation of the overall transmission to highlight the multiple degrees of freedom of each part.

3.2. An Insight on the Low Fidelity Model

As already mentioned, a Low Fidelity model (LF) has been developed, starting from the High Fidelity model.

The LF presents a very similar structure compared with the HF model (5): the various blocks share the same high level functions but they are simplified and lightened.

The LF sees inside a command generator, a controller module, an electromagnetic model module as well as a load and a dynamical model. The only main difference with respect to the HF model lies in the exploitation of an equivalent single-phase approach. The equivalent single phase is strictly related to the quadrature component reconstructed from the measured current flowing in the stator windings. The calculation utilise Clarke-Park transformation and a low pass filter.

Furthermore, some minor simplifications in each block can be found, such as a lightened PID controller, without any anti-windup or derivative filtering system. Finally the mechanical model considers a linear contribution of viscous effects.

This model has been validated thanks to the already experimentally validated HF model in a multi parameter process. Given the strong system-oriented characteristics of the problem, the adopted tuning strategy and performance validation metric has taken into consideration various aspects aimed at making the LF model (integrated in the prognostic algorithm) capable of simulating the real system with the desired behaviour: low computational time, medium to high accuracy and sensibility, satisfactorily low error with respect to the HF model. A more in depth description of the model and the relative validation and tuning process can be found in (Berri, 2021), (Berri et al., 2019) and (Berri et al., 2018).

4. RESULTS

User speed and position measurements have been acquired thanks to the external encoder, whereas motor speed and position measurements have been obtained thanks to the Control Unit software, through a resolver sensor integrated in the motor.

4.1. Low Fidelity and High fidelity

Figure 6 and 7 shows a very good behaviour of the LF model, which is able to simulate the commanded signal with minimum differences, despite the much lower computational cost and the much simpler modelling approach.

This confirms that a real-time monitoring system can be approached with low fidelity models trained with the help of more complex and computational intensive high fidelity ones. In particular, position and speed are almost identical. The LF result has been experimentally validated in nominal conditions and displays valid results.

4.2. Non-nominal conditions.

The installation of the external encoder support assembly, combined with the backlash modelling, has allowed us to experimentally validate the HF model in the presence of incipient mechanical faults.

In fact, variable backlash has been inserted at the output ring of the planetary gearbox.

A minor tuning of model controller gains and parameters has been carried out, comparing the data with the information given by Siemens software (e.g. proportional gain, error position saturation, current saturation) and with CAD evaluation (i.e. the planetary gearbox moment of inertia).

A first set of measurements and calibration has been carried out:

- the horizontal measurement read from the micrometer (d) (starting from $14mm$);
- the distance between the centers of the gears (Wheelbase distance MW);

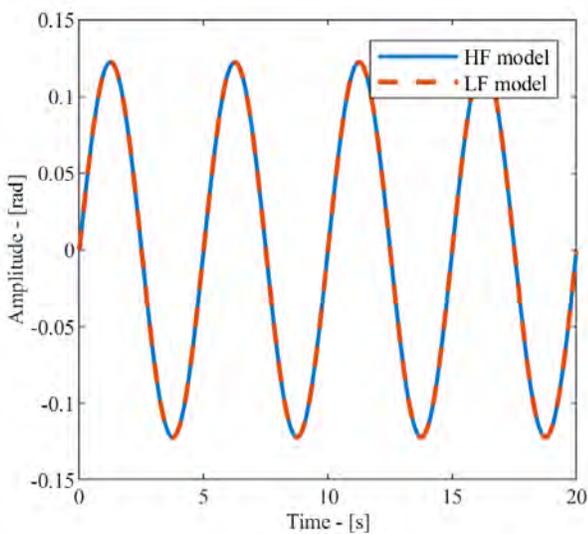


Figure 6. Comparison between LF and HF position predictions.

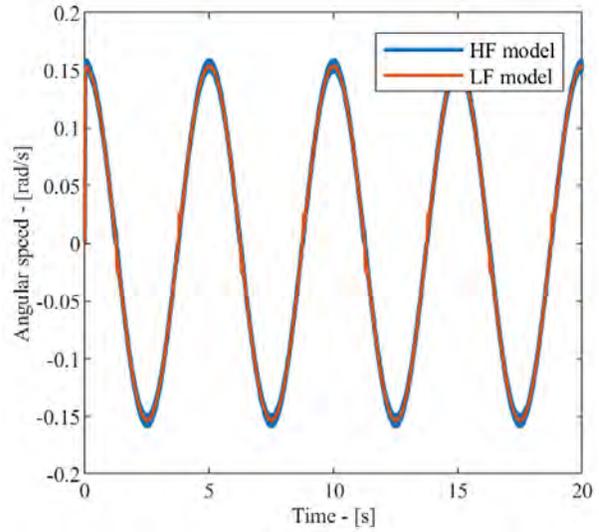


Figure 7. Comparison between LF and HF angular speed predictions.

- the radial spacing Δ_{RS} , which is defined as the wheelbase difference with respect to the zero backlash condition:

$$\Delta_{RS} = MW - MW_{ZeroBacklash} \quad (4)$$

Then a theoretical Backlash formula (taken from (Boggio, 2021)) has been used to estimate backlash amplitude:

$$Est.BLK[^{\circ}] = \frac{\Delta_{RS} \cdot \tan \phi}{z} \cdot m \cdot \frac{180^{\circ}}{\pi} \quad (5)$$

which refers to ideal teeth profiles. ϕ is the pressure angle (in this case 20°), m is the gear module (in this case 2) and z is the number of teeth of the output ring of the planetary gearbox.

A set of measurement has been completed and shown in Table 2, comparing the actual backlash read from the encoder with the one estimated using Eq. 5.

It has to be noted that the gearbox assembly as a whole shows an intrinsic and uncontrollable backlash due to mounting, coupling and manufacturing tolerances; all these contributions sum up and are considered to be equal to 0.3667° .

This is shown in Table 2, with the sixth column which reports the "net" backlash, that is the measured backlash (column Exp. BLK) minus the intrinsic (nominal) backlash. Therefore, the last two columns can be compared to show the adherence of the results obtained with theoretical formula with the experimental values.

Figure 8 shows that the values are indeed very similar. They tend to be different at higher radial spacing values (that is

lower horizontal distances) due to the gears involute curve tooth profile which is not precise but approximated due to the FDM manufacturing process.

This preliminary analysis has been useful to foresee reasonable backlash parameters and to limit the backlash amplitude to a certain acceptable range so that the "geometrical" calculation and the experimental measurement do not diverge in an excessive way.

The test campaign has been carried out with a position command with the following characteristics:

- Sinusoidal waveform;
- Amplitude: 7° ;
- Bias: 0°
- Frequency: $0.2Hz$.

Figure 9 shows good model behaviour in zero external backlash conditions (i.e. the first row of Table 2 with only nominal backlash). If we look at the highest point of the signal, it is possible to see the nominal backlash in action, both on the experimental signal and on the modelled one.

The amplitude of the experimental signal is slightly bigger than the modelled one. This can be traced back to minor imprecision inside the acquisition modules; the error has been proved to be constant as the gear backlash amplitude changes, hence minor tunings of internal gains will lessen the amplitude differences.

Comparisons between modelling and experimental results regarding no-load tests are shown in Figure 11 and Figure 10 where position and speed data are reported.

Three graphs are reported for position and speed as they refer to zero backlash condition (test no. 1 in Table 2), medium

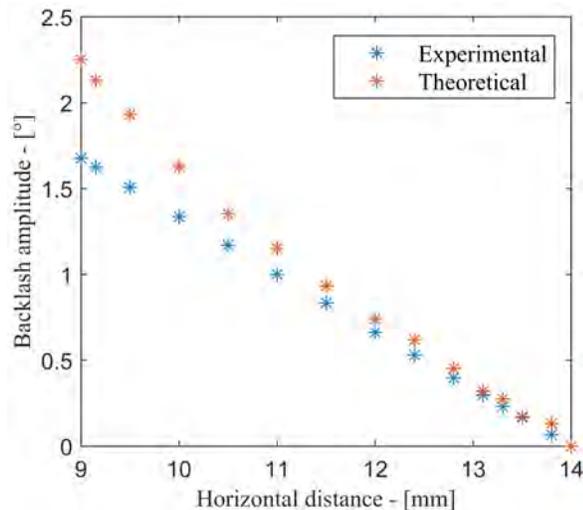


Figure 8. Comparison between theoretical and experimental backlash.

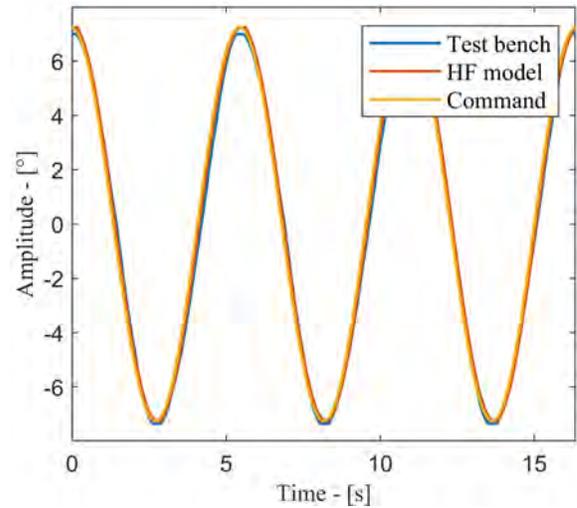


Figure 9. Commanded position, Simulink Model and Test Bench in zero backlash condition.

backlash condition (test no. 7 in Table 2) and maximum backlash condition between the gears (test no. 14 in Table 2). The graphs show promising results at steady states regimes, as the model is able to reproduce the intricate relationships between mechanical parts with a quite basic backlash modelling.

On the contrary, in transient states the model does not reproduce the effective trend efficiently, probably due to difficulties of modelling transient regimes, which are known to present multiple non linearities that have to be addressed separately. Hence this part is not shown.

Nonetheless, the model is able to distinguish between moments where the gears are in contact with each others and moments where there is no contact at all.

In addition, as can be noted in Figure 10 concerning speed comparisons, the model can successfully predict even small spikes, proving the goodness of the model design and confirming the validity of the overall model.

Position comparison shows excellent forecasting, especially at steady state, even in presence of maximum backlash conditions: at the highest and lowest value in the sinusoidal motion the model manages to deliver the typical horizontal line with the right time length.

In those conditions there is no contact between the gears. Therefore, even if the command is transferred to the motor and to the gearbox, the encoder gear still does not feel any movement. Only when the gap is closed, the motion is transmitted to the gear.

As reported in (Maré, 2017), this is one of backlash most detrimental effects, since there is a delay in the transmission and the user gear is free to move.

In the steeper part of the sinusoidal motion, the position is not influenced by backlash at all, since the gears always move in the same direction.

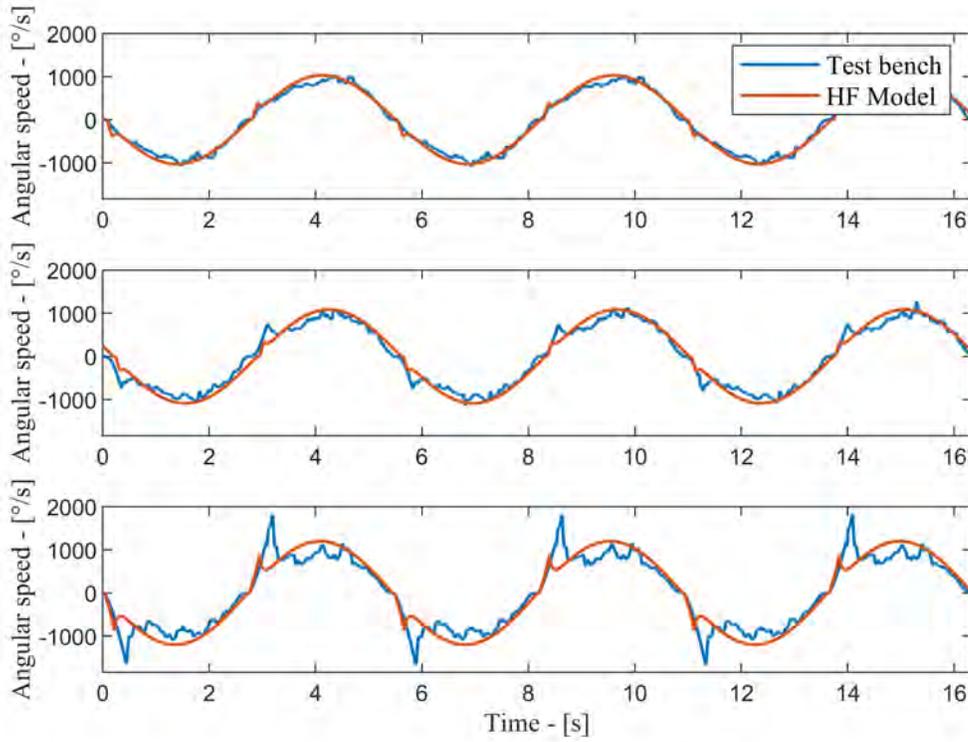


Figure 10. Model and experimental speed data in different backlash conditions (14mm – 12.4mm – 9.15mm).

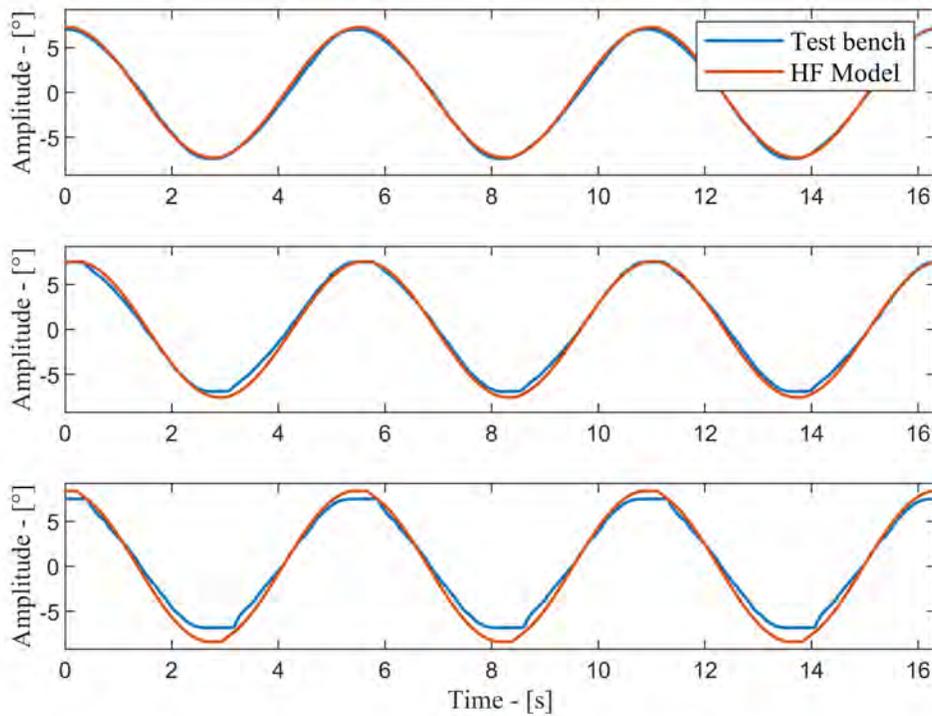


Figure 11. Model and experimental position data in different backlash conditions (14mm – 12.4mm – 9.15mm).

Table 2. Estimated and experimental backlash.

-	d	MW	Δ_{RS}	Exp. BLK	Net BLK	Est. BLK
1	14	108.43	0	0.3667	0	0
2	13.8	108.57	0.14	0.4985	0.1318	0.0664
3	13.5	108.78	0.35	0.5386	0.1719	0.1653
...						
7	12.4	109.55	1.12	0.9855	0.6188	0.5308
8	12	109.83	1.4	1.1058	0.7391	0.6635
9	11.5	110.19	1.76	1.3006	0.9339	0.8342
...						
13	9.5	111.61	3.18	2.2976	1.9309	1.5072
14	9.15	111.86	3.43	2.4981	2.1314	1.6257
15	9	111.97	3.54	2.6184	2.2517	1.6778

5. CONCLUSION

A new numerical HF model, which is able to simulate successfully backlash conditions, has been developed by the authors and presented showing remarkable ability to foresee trends in position and speed change at steady-states regimes. In fact, the model has proved significant ability to simulate the complex contact dynamics between gears and to distinguish between contact and non-contact integration steps. Furthermore, even small spikes and minor disturbances are reproduced as well.

These trends and the model itself has been validated using an updated test bench designed and assembled by the authors. The newly developed model can be further modified and improved involving more accurate backlash modelling which might be able to track even transient regimes of speed and position.

Moreover, a parallel low fidelity model has been refined. The comparison between the two models shows a good trend overlap, highlighting the quality of the LF model.

Finally, additional validation and comparisons will be carried out with variable friction, exploiting the load module on the test bench: this will further verify the model goodness in presence of strong non linearities. Real-life applications for this model are endless, especially if applied on Low Fidelity models which could run in real-time. The real implementation of these techniques is deemed to be cost-effective, as the computational burden is reduced and necessary components (e.g. sensors) and technologies are already largely widespread in the aerospace sector.

ACKNOWLEDGMENT

The authors wish to thank Matteo Bertone and Luca Boggio for the work carried out during their master theses.

NOMENCLATURE

I_{ref}	Reference current value
i_j	Actual current in the j -th phase
k_j	Back-EMF coefficient for the j -th phase
T_m	Motor torque
T_l	External torque (load)
J_m	Actuator's rotating assembly inertia
C_m	Actuator's rotating assembly viscous friction coefficient
θ_m	Motor mechanical position (gearbox upstream input)
BLK	Backlash amplitude
θ_u	User position (gearbox downstream output)
t	Simulation time
dt	Simulation step time
MW	Wheelbase distance
d	Horizontal measurement read from the micrometer
Δ_{rs}	Radial spacing
$Est.BLK$	Estimated theoretical backlash amplitude
ϕ	Pressure angle
m	Gear module
z	Number of teeth
$Exp.BLK$	Experimental backlash amplitude
$Net.BLK$	Net backlash amplitude, without intrinsic backlash

REFERENCES

- Abdelhafez, A., & Forsyth, A. (2009). A review of more-electric aircraft. In *International conference on aerospace sciences and aviation technology* (Vol. 13, pp. 1–13).
- Balaban, E., Bansal, P., Stoelting, P., Saxena, A., Goebel, K. F., & Curran, S. (2009). A diagnostic approach for electro-mechanical actuators in aerospace systems. In *2009 IEEE Aerospace Conference* (pp. 1–13).
- Baldo, L. (2021). Development of an experimental test bench for the validation of prognostic algorithms for electromechanical actuators. *MSc Thesis*.

- Berri, P. C. (2021). Design and development of algorithms and technologies applied to prognostics of aerospace systems. <https://iris.polito.it/handle/11583/2927464>.
- Berri, P. C., Dalla Vedova, M. D., & Mainini, L. (2021). Computational framework for real-time diagnostics and prognostics of aircraft actuation systems. *Computers in Industry*, 132, 103523.
- Berri, P. C., Dalla Vedova, M. D., & Mainini, L. (2022). Learning for predictions: Real-time reliability assessment of aerospace systems. *AIAA Journal*, 60(2), 566–577.
- Berri, P. C., Dalla Vedova, M. D. L., Maggiore, P., & Riva, G. (June 2020). Design and development of a planetary gearbox for electromechanical actuator test bench through additive manufacturing. In *Actuators* (Vol. 9, pp. Issue 2, Article number 35. ISSN: 20760825 - DOI:10.3390/ACT902035 - Scopus: 2-s2.0-85085740653).
- Bertolino, A. C., De Martin, A., Jacazio, G., & Sorli, M. (2020). Towards a phm system for electro-mechanical flight control actuators. *I-RIM 2020*, 4.
- Boggio, L. (2021). Development of an experimental test bench for the validation of prognostic algorithms for electromechanical actuators. *MSc Thesis*.
- Borello, L., & Dalla Vedova, M. D. L. (July 2006). Load dependent coulomb friction: a mathematical and computational model for dynamic simulation in mechanical and aeronautical fields. *International Journal of Mechanics and Control (JoMaC)*, Vol. 7(No. 1), pp. 19–30.
- Boschetti, V. (2020). Development of an experimental test bench for the validation of prognostic algorithms for electromechanical actuators. *MSc Thesis*.
- De Martin, A., Jacazio, G., & Vachtsevanos, G. (2017). Windings fault detection and prognosis in electromechanical flight control actuators operating in active-active configuration. *International Journal of Prognostics and Health Management*, 8(2).
- Di Rito, G., Denti, E., & Galatolo, R. (2008). Development and experimental validation of real-time executable models of primary fly-by-wire actuators. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 222(6), 523–542.
- Garcia, P. L., Crispel, S., Verstraten, T., Saerens, E., Vanderborght, B., & Lefeber, D. (2019). Wolfrom planetary gear trains for lightweight, human-centered robotics. In *International conference on gears 2019* (pp. 753–764).
- Garcia Garriga, A., Ponnusamy, S. S., & Mainini, L. (2018). A multi-fidelity framework to support the design of more-electric actuation. In *2018 multidisciplinary analysis and optimization conference* (p. 3741).
- Giangrande, P., Madonna, V., Sala, G., Kladas, A., Gerada, C., & Galea, M. (2018). Design and testing of pmsm for aerospace ema applications. In *Iecon 2018-44th annual conference of the ieee industrial electronics society* (pp. 2038–2043).
- Mainini, L., & Willcox, K. (2015). Surrogate modeling approach to support real-time structural assessment and decision making. *AIAA Journal*, 53(6), 1612–1626.
- Maré, J.-C. (2017). *Aerospace actuators 2: signal-by-wire and power-by-wire* (Vol. 2). John Wiley & Sons.
- Nordin, M., & Gutman, P.-O. (2002). Controlling mechanical systems with backlash—a survey. *Automatica*, 38(10), 1633–1649.
- Qiao, G., Liu, G., Shi, Z., Wang, Y., Ma, S., & Lim, T. C. (2018). A review of electromechanical actuators for more/all electric aircraft systems. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 232(22), 4128–4151.
- Sciandra, P. (2020). Development and experimental validation of prognostic algorithms for electromechanical actuators. *MSc Thesis*.
- Sutharssan, T., Stoyanov, S., Bailey, C., & Yin, C. (2015). Prognostic and health management for engineering systems: a review of the data-driven approach and algorithms. *The Journal of engineering*, 2015(7), 215–222.
- Williams, Z. (2006). Benefits of ivhm: an analytical approach. In *2006 ieee aerospace conference* (pp. 9–pp).

An Analysis of Vibrations and Currents for Broken Rotor Bar Detection in Three-phase Induction Motors

Amirhossein Berenji¹, Zahra Taghiyarrenani²

¹ *Department of Mechanical and Energy Engineering, Shahid Beheshti University, Tehran, Tehran, 1983969411, Iran
a.berenji@mail.sbu.ac.ir*

² *Center for Applied Intelligence Systems Research, Halmstad University, Halmstad, Halland, 30118, Sweden
zahra.taghiyarrenani@hh.se*

ABSTRACT

Selecting the physical property capable of representing the health state of a machine is an important step in designing fault detection systems. In addition, variation of the loading condition is a challenge in deploying an industrial predictive maintenance solution. The robustness of the physical properties to variations in loading conditions is, therefore, an important consideration. In this paper, we focus specifically on squirrel cage induction motors and analyze the capabilities of three-phase current and five vibration signals acquired from different locations of the motor for the detection of Broken Rotor Bar generated in different loads. In particular, we examine the mentioned signals in relation to the performance of classifiers trained with them. Regarding the classifiers, we employ deep conventional classifiers and also propose a hybrid classifier that utilizes contrastive loss in order to mitigate the effect of different variations. The analysis shows that vibration signals are more robust under varying load conditions. Furthermore, the proposed hybrid classifier outperforms conventional classifiers and is able to achieve an accuracy of 90.96% when using current signals and 97.69% when using vibration signals.

1. INTRODUCTION

Being the origin of motion, electric motors play a vital role in rotary systems. Due to the ease of operation, affordability, and structural simplicity of induction motors, they are the most commonly used type of electric motor in the industry (Tsyppkin, 2017; Kanović et al., 2013). Rotors in induction motors are manufactured to be quite robust nowadays, but there are still various faults expected, including Broken Rotor Bar (BRB) (Kanović et al., 2013). BRB faults share same starting stage, where there is simple crack in the rotor bar

(Ferrucho-Alvarez et al., 2021). In case this fault is not diagnosed and the essential corrective actions are not taken, BRB with serious severity and probably other faults are unavoidable (Wang et al., 2019).

In addition, the essentially of simultaneously low cost and reliable production has resulted in a paradigm shift in rotating machinery maintenance strategy, from corrective to preventive maintenance (Yan, Gao, & Chen, 2014). One of the preventative maintenance methods that has gained increasing attention in recent years is data-driven methods. In order to provide data for such methods, different physical properties may be utilized. As current and vibration signals are two of the most commonly used properties for BRB detection (Gritli et al., 2012), it is crucial to understand how these two signals can be used to detect the BRB from a data-driven perspective.

Furthermore, the induction motors, in general, have the advantage of being able to operate under variable loads (Sonowal, Gogoi, Boruah, & Barman, 2019); however, this feature poses a challenge to data-driven methods. This challenge originates in the fact that every variation of loading condition would also vary the dynamics of the machine; resulting in different sample distributions, which adversely affect the performance of a data-driven model (Sonowal et al., 2019). Therefore, when constructing a data-driven model, it is important to take into account load variations. This subject comes to higher level of importance regarding the BRB diagnosis, where most approaches require the operation of the motor on heavy load (Ferrucho-Alvarez et al., 2021).

In this paper, we analyze the current and vibration signals to detect BRBs. To accomplish this, we compare the accuracy of fault prediction models trained on current and vibration signals. By taking into account the various load variations, we develop a classifier to mitigate these changes; subsequently, we evaluate the performance of the classifiers trained on the current and vibration signals. This evaluation enables us to compare the effectiveness of current and vibration signals to

Amirhossein Berenji et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

detect BRBs.

2. BACKGROUND

2.1. Contrastive learning and Siamese Neural networks

Contrastive Learning focuses on set of learning strategies and techniques involving learning by the comparing available samples and through their similarities and differences (Le-Khac, Healy, & Smeaton, 2020). These methods are great approaches to take, specifically in problems that construction of a feature space with noticeable separation of classes over the feature space. Siamese Neural Networks can be used to employ Contrastive Learning to extract such a feature set. A Siamese network consists of a pair of absolutely identical feature extractors that are supposed to derive embedding corresponding to an arbitrary pair of inputs. The network is trained in a manner which pairs with samples from dissimilar classes (negative pairs) orient far apart from each other, while pairs with samples belonging to similar classes (positive pairs) are mapped most closely to each other. Referenced training process would result in a fairly separable feature space in the embedding provided by the feature extractor.

Contrastive Loss can be used to train a siamese network. Its mathematical definition can be seen in the Equation 1. In this equation, Y is the label of a given pair (0 for negative pairs and 1 for positive pairs), D_w is a similarity index describing the similarity between the embeddings of the samples present in the pair and m is parameter known as margin. This function is consisted of two terms; the first term is supposed to represent observations of similar classes as closely possible, while the second term is responsible to increase the dissimilarity of the observations from different classes up to the highest extent (Jadon, 2020).

$$ContrastiveLoss = (1-Y)\frac{1}{2}D_w^2 + (Y)\frac{1}{2}(max(0, m-D_w))^2 \quad (1)$$

3. RELATED WORKS

Taking advantage of intelligent methods to analyze the motor vibrations for BRB detection is a well established approach and various studies can be found regarding this matter. For example, in (Su, Chong, & Ravi Kumar, 2011) Artificial Neural Networks are employed to implement an induction motor fault detection system, by analyzing vibrations of the machine. Similarly, in (Khan, Kim, & Choo, 2020) Dilated Convolutional Neural Networks are used to detect bearing faults in induction motors. In (Sadoughi, Ebrahimi, Moalem, & Sadri, 2007), Artificial Neural Networks and set of features derived from frequency spectrum of vibrations are used to detect the BRB problem in induction motors.

Intelligent methods are widely used in the study of induction

motor current signals for fault detection purposes too. For instance, in (Godoy, da Silva, Goedtel, Palácios, & Lopes, 2016), various intelligent methods including Artificial Neural Networks and Support Vector Machines are employed to both detect and classify the broken rotor bars in a three-phase induction motor. In (Bessam, Menacer, Boumehraz, & Cherif, 2016), a BRB diagnosis approach is proposed where Hilbert transform is used to extract features from stator current envelope; extracted features are then fed to a Multi-layered Perceptron to report the number of broken rotor bars, from zero to two. In (Valtierra-Rodriguez et al., 2020), short-time Fourier transform derives a time-frequency representation from motor current signals through its startup and Convolutional Neural Networks are employed to detect BRB problem.

4. COMPARISON OF CURRENTS AND VIBRATIONS FOR BRB DETECTION

In this section, we discuss the method that we use to compare the three-phase currents and vibrations, as describing modalities of BRB problem in induction motors. We aim to evaluate the separability of the different induction motor health classes from BRB point of view (including no broken rotor bar to four broken rotor bars), based on current and vibrations. Therefore, we can determine which modality is more effective for detecting BRB. To this end, we employ conventional deep neural networks initially; We train two multi-layer perception neural networks for BRB Detection using current and vibration signals, respectively. The results of the evaluation describes the separability of different health classes in each modality. The reason is that the classifiers are unable to detect samples that belong to different classes but overlap with each other. However, the overlapping of different classes can be because of the variations in loads. Therefore, we design a hybrid classification method that is able to compensate for load variations; Using this method, we are able to compare vibration and current signals after the effects of variations in load have been eliminated.

The figure 1 illustrates the proposed hybrid classification method in order to compensate for load variations. At first, we pair samples from different loads. Two paired samples taken from the same classes, regardless of the loads, are considered as a positive pair and two samples taken from different classes, regardless of the loads, are considered as a negative pair. Using the positive and negative pairs called *training pairs*, we train a Siamese neural network with Contrastive Loss. As a result of this training, a function as feature extractor called FE is generated. The FE maps the samples to a new embedding space. In the embedding space, positive pairs will be placed close together. It means that the samples with same classes, regardless of their loads, will be grouped. Additionally, the negative pairs will be distant from each other. As a result, different loads will be aggregated in this embedding. We then add a softmax layer to the FE and retrain it using the

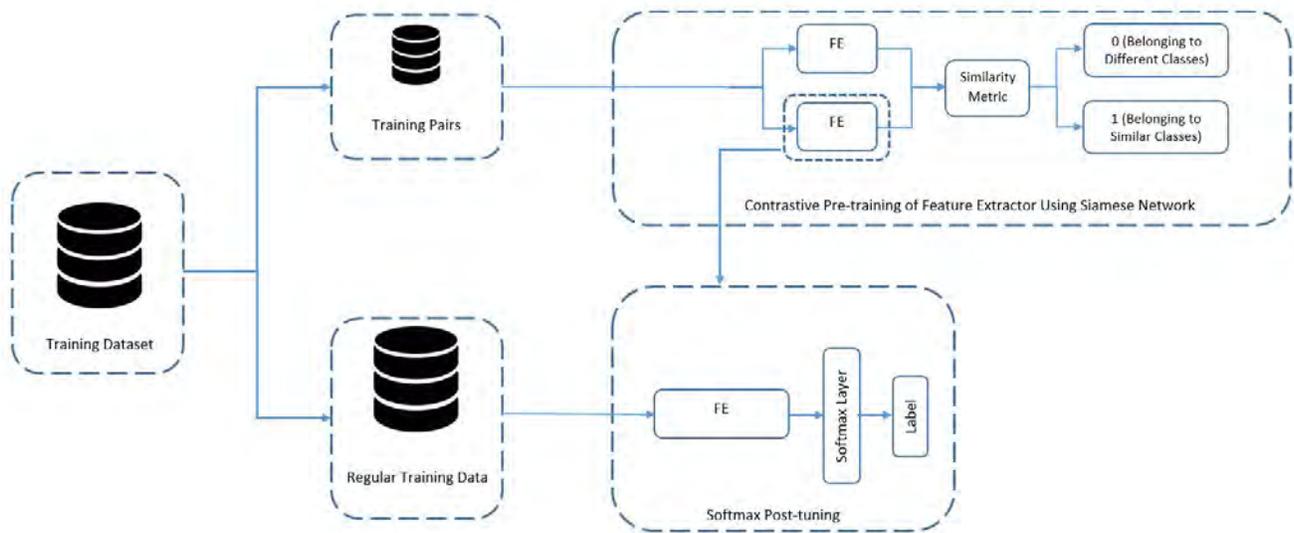


Figure 1. Visual demonstration of the Hybrid Classification Approach

available samples, similar to a conventional classifier. Using the current and vibration signals we train two classifiers using hybrid approach and evaluate the results for the purpose of comparing current and vibration signals.

5. EXPERIMENTS

5.1. Dataset

The experimental dataset for detecting and diagnosing rotor broken bar in a three-phase induction motor is used to conduct the comparative study that this paper aims to carry out (Tremblay, Flauzino, Suetake, & Maciejewski, 2020). This dataset includes three-phase voltages, three-phase currents and vibrations signals, collected from various locations of the motor, in various loading conditions. In addition to healthy operation of motor cases involving 1 to 4 broken rotor bars are included in this dataset. Moreover, the dataset contains eight levels of mechanical torques as loading conditions, from 12.5% to 100% of nominal load (4 N.m), to evaluate the effect of load variation. In our study, we took advantage of four load levels, including 12.5%, 50%, 62.5% and 100% of nominal load.

5.2. Data Pre-processing and Preparation Procedure

Original time domain signals in both current and vibration modalities are split to time domain signals with lengths of respectively 6667 and 1024 points long signals. Consecutively, Fast Fourier Transform is employed to alter the time domain observations to frequency domain records, as BRB is easier to detect in frequency domain. For each loading condition referenced previously, the training and testing splitting

process is done using random selection. Test size of 25% is employed. Random states are preserved to assist the reproducibility of results. Moreover, the load-specific training splits are summed up to make the mixed-load training split. The mixed-load testing split can also be summed up, similarly. Feature scaling, as an important step of data pre-processing is done, using Min/Max scaling.

5.3. BRB Detection using Conventional Deep Classifiers

The first set of experiments conducted on this dataset involves training deep classifiers on both current and vibration modalities, separately. The classification problem to be solved involves detecting the number of broken rotor bars, from zero to four, given either three-phase current signals or vibrations signals. Due to the difference in the size of concatenated three-phase currents signals and its vibration counterparts, networks used for each modality is different from the other. In Table 1, the size of each network is included. Except for the last layer in each network, which is supposed to be a Softmax layer in multi-class classification problems, rest of the layers in both networks employ Hyperbolic Tangent as the activation function. As a conventional loss function for multi-class classification problems, Categorical Cross-entropy is used as the loss function to train classification networks. Moreover, the Adam optimizer is used to minimize the loss function during the training process, where the learning rate is chosen to be 0.000001 and decay is fixed as the division of learning rate by number of epochs. For the sake of training both currents and vibrations, 400 epochs provided well-stabilized training procedure, therefore same value used for both of them.

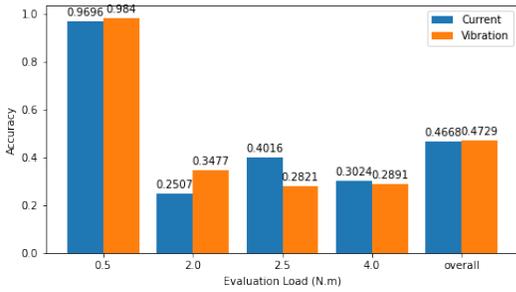


Figure 2. The results of a conventional classifier trained with the samples from load 0.5

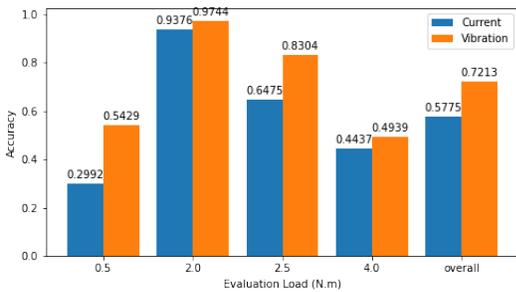


Figure 3. The results of a conventional classifier trained with the samples from load 2.0

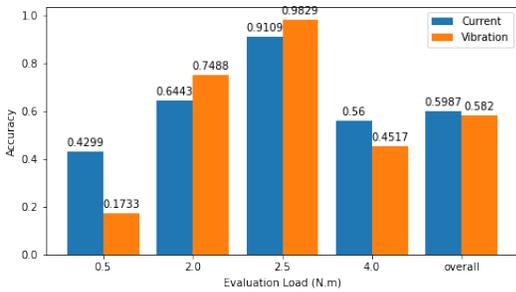


Figure 4. The results of a conventional classifier trained with the samples from load 2.5

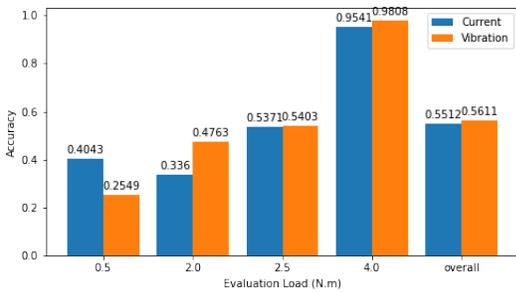


Figure 5. The results of a conventional classifier trained with the samples from load 4.0

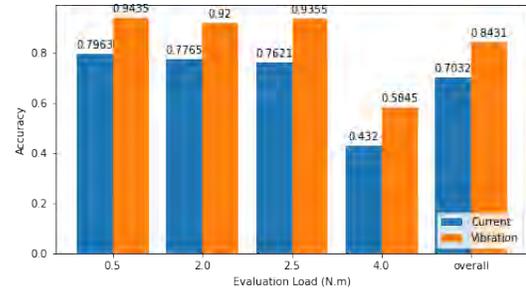


Figure 6. The results of conventional classifiers trained with the current and vibration samples from all load

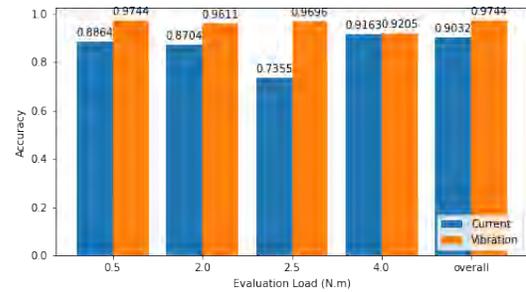


Figure 7. The results of hybrid classifier trained with the the current and vibration samples from all load

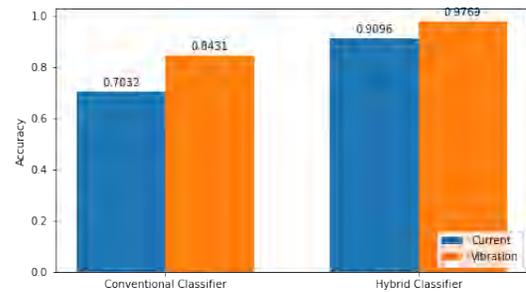


Figure 8. The effectiveness of contrastive pre-training in hybrid-classification approach on the improvement of mean classification accuracies in mixed-load scenarios

The generalizability of what is learnt by classifiers on each load over different loads is quantified by evaluating its performance over not only the training load, but other three loads and the mixture of them. Results obtained by three-phase currents classifiers and vibrations classifiers are gathered in the Figures 2 to 6. To compensate for the effect of randomness, experiments in this section are repeated over 5 trials and the mean classification accuracy is summarized. The figure 2 shows the results of a conventional classifiers that is trained using the samples from load 0.5. Likewise, figures 3 to 6 show the results of the classifiers that are trained using the samples from load 2.0, 2.5, 4.0, and mixture of all loads, respectively. Based on the results provided, it is clearly understood that both modalities perform acceptable in solving the classification problems, when the evaluation load is identical

Table 1. The Structure of Network for each Modality

Modality to be used	Neurons per Layer
Three-phase Currents	9999-7500-6000-4500-3000-1500-750-500-250-50-5
Vibrations	2560-1280-640-580-512-256-128-64-5

to the training load, however, both modalities experience severe decrease in the classification accuracy when classifiers are evaluated on loads, rather than training load. It is also obvious that in most cases, the farther evaluation load is from training load, the more drastic would be the referenced decrease. In addition to those, vibrations offers higher classification performance in mixed-load scenario, in comparison with the currents.

5.4. Hybrid Classification Approach to Overcome Data Drift due to Load Variation

Based on the results available in the Figures 2 to 5, conventional classifiers fail to perform well in mixed load scenarios, no matter which modality is used. Therefore, we evaluate the effectiveness of the proposed hybrid classifier; it means we evaluate the effectiveness of a Contrastive Representation Learning pre-training step to make the feature extraction section(classification network, excluding the softmax layer at its end) of the classification networks, to derive a more robust feature space to load variation. To this end, we use 25% of the training data available for this step. Networks employed in this section follow the exact same architecture of the networks, discussed in the previous section. Number of positive and negative pairs used during the pre-training is kept equal to preserve the pre-training step a balanced training process. Moreover, the number of pairs per each observation in training set used during the pre-training are found by increasing the number of pairs until there is no significant improvement in validation accuracy by increasing the number of pairs, in which 10 and 4 were found as optimum number of pairs for vibrations and currents, respectively. The loss function employed during the pre-training is Contrastive Loss and Adam optimizer is used as the optimizer. Learning rate is fixed at 0.00001 and 100 epochs provided sufficient iterations of training process. Similar to previous experiments, the division of learning rate by number of epochs, is used as the decay parameter of the optimizer. Afterwards to the Contrastive Representation Learning pre-training, a softmax layer is added to the feature extractor and the whole network (feature extractor and the softmax layer) is post-trained, using the remaining 75% of the training data. The post-training procedure of the whole network involving the addition of softmax layer and retraining of the whole network, employs exactly the same set of parameters used during the previous experiments. Similar to the results from previous experiments, these experiments are conducted over 5 trials to exclude the effect

of randomness through training process.

According to the Figure 7, contrastive pre-training improves the classification performance significantly for all loads. In addition, the obtained accuracies per load are almost identical; it means that the hybrid classifier is able to aggregate the different loads and consequently makes higher levels of classification accuracy achievable. In addition, similar to the previous set of experiments, still vibrations outperforms currents in the classification performance. To be able to compare the the vibration and current signals using both conventional and hybrid classifiers, figure 8 summarizes the results. We can clearly see that vibration signals can be more effective for detecting the BRBs.

6. CONCLUSION

This paper studies the robustness of currents and vibrations towards mechanical load variation, for Broken Rotor Bar problem detection in squirrel cage induction motors. Our experiments proved that vibrations is less sensitive towards mechanical load variation. Moreover, we assessed the effectiveness of a contrastive representation learning pre-training in the reconstruction of a feature set in which data drift due to load variation is compensated. Contrastive learning-based pre-training offered significant improvement in the classification accuracy in both modalities. The superiority of vibrations over current in BRB detection is still noticeable, even afterwards of the employment of the pre-training step. Comparison of the robustness of current and vibrations towards mechanical load variation in the detection of other faults of induction motors can be considered as the subject of future work.

REFERENCES

- Bessam, B., Menacer, A., Boumehraz, M., & Cherif, H. (2016). Detection of broken rotor bar faults in induction motor at low load using neural network. *ISA transactions*, 64, 241–246.
- Ferrucho-Alvarez, E. R., Martinez-Herrera, A. L., Cabal-Yeppez, E., Rodriguez-Donate, C., Lopez-Ramirez, M., & Mata-Chavez, R. I. (2021). Broken rotor bar detection in induction motors through contrast estimation. *Sensors*, 21(22), 7446.
- Godoy, W. F., da Silva, I. N., Goedel, A., Palácios, R. H. C., & Lopes, T. D. (2016). Application of intelligent tools to detect and classify broken rotor bars in three-phase induction motors fed by an inverter. *IET Electric Power Applications*, 10(5), 430–439.
- Gritli, Y., Di Tommaso, A., Filippetti, F., Miceli, R., Rossi, C., & Chatti, A. (2012). Investigation of motor current signature and vibration analysis for diagnosing rotor broken bars in double cage induction motors. In *International symposium on power electronics power*

- electronics, electrical drives, automation and motion* (pp. 1360–1365).
- Jadon, S. (2020). An overview of deep learning architectures in few-shot learning domain. *arXiv preprint arXiv:2008.06365*.
- Kanović, Ž., Matic, D., Jeličić, Z., Rapaić, M., Jakovljević, B., & Kapetina, M. (2013). Induction motor broken rotor bar detection using vibration analysis—a case study. In *2013 9th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives (SDMPED)* (pp. 64–68).
- Khan, M. A., Kim, Y.-H., & Choo, J. (2020). Intelligent fault detection using raw vibration signals via dilated convolutional neural networks. *The Journal of Supercomputing*, 76(10), 8086–8100.
- Le-Khac, P. H., Healy, G., & Smeaton, A. F. (2020). Contrastive representation learning: A framework and review. *IEEE Access*, 8, 193907–193934.
- Sadoughi, A., Ebrahimi, M., Moalem, M., & Sadri, S. (2007). Intelligent diagnosis of broken bars in induction motors based on new features in vibration spectrum. In *2007 IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives* (pp. 106–111).
- Sonowal, M., Gogoi, B. B., Boruah, M., & Barman, J. K. (2019). Health monitoring of induction motor through vibration analysis. *ADBU Journal of Electrical and Electronics Engineering (AJEEE)*, 3(1), 1–8.
- Su, H., Chong, K. T., & Ravi Kumar, R. (2011). Vibration signal analysis for electrical fault detection of induction machine using neural networks. *Neural Computing and Applications*, 20(2), 183–194.
- Treml, A. E., Flauzino, R. A., Suetake, M., & Maciejewski, N. A. R. (2020). Experimental database for detecting and diagnosing rotor broken bar in a three-phase induction motor. *IEEE DataPort*.
- Tsyppkin, M. (2017). Induction motor condition monitoring: Vibration analysis technique—diagnosis of electromagnetic anomalies. In *2017 IEEE Autotestcon* (pp. 1–7).
- Valtierra-Rodriguez, M., Rivera-Guillen, J. R., Basurto-Hurtado, J. A., De-Santiago-Perez, J. J., Granados-Lieberman, D., & Amezcua-Sanchez, J. P. (2020). Convolutional neural network and motor current signature analysis during the transient state for detection of broken rotor bars in induction motors. *sensors*, 20(13), 3721.
- Wang, Z., Yang, J., Li, H., Zhen, D., Xu, Y., & Gu, F. (2019). Fault identification of broken rotor bars in induction motors using an improved cyclic modulation spectral analysis. *Energies*, 12(17), 3279.
- Yan, R., Gao, R. X., & Chen, X. (2014). Wavelets for fault diagnosis of rotary machines: A review with applications. *Signal processing*, 96, 1–15.

Online Flow Estimation for Condition Monitoring of Pumps in Aircraft Hydraulics

Phillip Bischof¹, Frank Thielecke², and Dirk Metzler³

^{1,2} *Hamburg University of Technology, Institute of Aircraft Systems Engineering, Neßpriel 5, 21129 Hamburg, Germany*
 phillip.bischof@tuhh.de
 frank.thielecke@tuhh.de

³ *Liebherr-Aerospace Lindenberg GmbH, Pfänderstraße 50-52, 88161 Lindenberg/Allgäu, Germany*
 dirk.metzler@liebherr.com

ABSTRACT

Hydraulic systems in conventional civil aviation are currently monitored in a very rudimentary way. Normally, measured values are compared with a fixed threshold. If these measured values are outside the predefined limits, the entire hydraulic system is usually shut down. To overcome this deficit, a study regarding a novel prognostic health management method for aircraft hydraulic pumps, which allows a statement about the pump condition, is presented in this paper. The method is based on measuring differential pressure and temperature at a suitable resistance. In the first part of the study, the overall concept for monitoring the motor pump unit is analyzed. This is followed by a discussion of possible measurement methods and suitable resistors to determine the condition of the pump. In the second part of the study, the implementation for online monitoring of the pump is discussed. After a suitable approximation is found, the quality of the proposed method is evaluated with real hydraulic power generation and consumers.

1. INTRODUCTION

Hydraulics play an essential role as a power supply in today's (modern) civil aviation. It can be assumed that due to increasing electrification, new electrohydraulic (eH) systems with high availability will be responsible for the actuation of various aircraft actuators, in form of highly efficient power packages (eHEPP). A simplified illustration of an eHEPP is presented in Figure 1. The eHEPP consists basically of two redundant Electric Motor Pumps (EMPs) and all other relevant components for hydraulic power generation e.g. filters, check valves and manifold (not depicted). A description gives (Trochelmann, Rave, Thielecke, & Metzler, 2017).

Phillip Bischof et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Each EMP includes a Motor Control Electronics Unit (MCE) and a Motor Pump Unit (MPU).

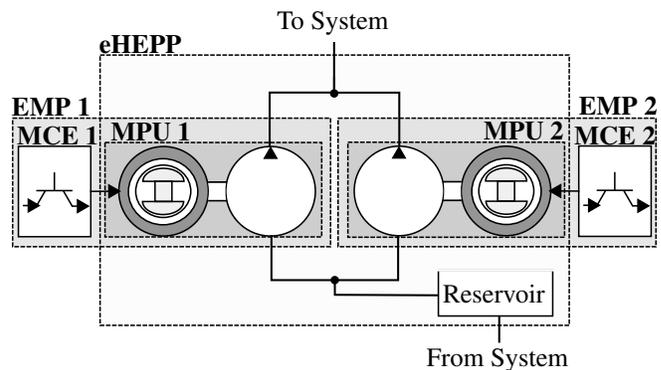


Figure 1. Simplified representation of the eHEPP

As the hydraulic energy generator, the pump is one of the most important components in the system and thus has a significant influence on the availability of the eHEPP, the entire hydraulic system and the flight controls. Current monitoring of EMPs in commercial aircraft is usually carried out by comparing actual values with limit values and does not allow any statement to be made about the condition of the pump. For example, the system pressure or the output pressure of the EMP is monitored by means of a pressure sensor or pressure switch (Poole, 2015). If a fault is detected, usually the entire hydraulic circuit is shut down, which can lead to flight cancellations and higher costs for the airline. To achieve an improvement in reliability, availability and ultimately a reduction in operating costs, it is necessary to implement an enhancement in monitoring of this critical element via Prognostic Health Monitoring (PHM). A possible PHM for aircraft hydraulic pumps is presented in this paper, which is structured as follows.

Section 2 introduces the considered MPU and its overall PHM concept. A brief introduction, assessment and selection of volumetric flow measurement methods is presented in Section 3. The theory behind the selected method is shown in Section 4. The implementation of the method follows in Section 5. Section 6 includes some lessons learned and the conclusion of this study is found in Section 7.

2. HEALTH MONITORING OF THE MPU

The following section discusses the EMP of the eHEPP in more detail. The operating principle of this device will also be explained. In addition, the basic concept for the prognostics and health monitoring of the hydraulic pump is described.

2.1. The Motor-Pump-Unit

Constant pressure hydraulic systems are state of the art in aircraft hydraulics. Typically, hydraulic power is supplied by a pressure compensated axial piston pump (AKP). The pump is driven by the aircraft engine (Engine Driven Pump - EDP) or an electric motor (EMP). The motor of this state of the art EMP (e.g. in the Airbus A320) is supplied with constant frequency current, thus the pump rotates with constant speed. The pressure is then regulated to the desired level with the swashplate of the AKP. This EMP is usually called the fixed speed variable displacement - EMP (FSVD-EMP). In comparison, the EMP presented in this study controls the system pressure by changing the pump speed. Figure 2 shows a simplified representation of the electric motor driven pump prototype.

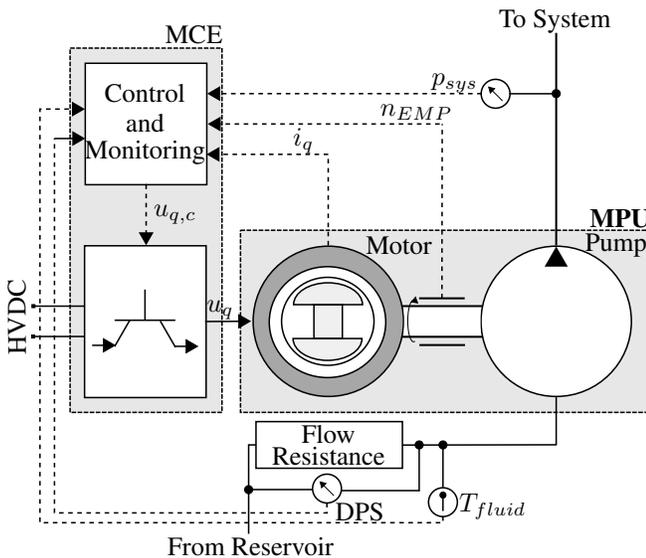


Figure 2. Simplified representation of the VSFD-EMP

The prototype is a variable speed fixed displacement (VSFD)-EMP and essentially consists of three main components. The first component is the motor control electronics (MCE), which

in turn consists of a control and monitoring unit and an inverter. The second component is the electric motor, in this case a permanent magnet synchronous motor (PMSM). Finally, the last component is the hydraulic pump, more specifically an internal gear pump (IGP).

As already described the system pressure is regulated by adjusting the pump speed. A baseline pressure control concept is described in (Trochermann, Bischof, Thielecke, Metzler, & Bassett, 2018). It is a common cascade control concept. In the innermost loop of the control system current is controlled. The speed controller is located in the middle loop. Lastly, the system pressure is controlled in the outer loop. The main measurements and signals needed for pressure control with the EMP prototype are also depicted in Figure 2.

Since the VSFD-EMP must be controlled digitally, the current, speed, and pressure of the system are known. Compared to the state of the art EMP, the Fixed Speed Variable Displacement (FSVD)-EMP, the new concept provides substantially more information than the FSVD-EMP type, especially the MPU speed and its known fixed displacement. Therefore, the new EMP prototype enables new concepts for monitoring the pump condition. This concept is presented next.

2.2. PHM Concept for Motor-Pump-Unit

As previously shown, the HePP includes two redundant MPUs so that the required availability can be met. This does not always mean that both EMPs are active at the same time. At this stage of development, the proposed health monitoring concept assumes that only one EMP is active, though this consideration is heavily dependent of the design of the system. This postulation does not lead to the notion that it is always the same EMP that is active. The active EMP should be changed continuously, e.g., after each flight. This not only reduces the possibility of highly uneven degradation of the MPUs, but also allows health monitoring each second flight which minimizes dormant times.

In general, degradation, which affects performance, of MPUs can be divided into two main categories, (hydro-)mechanical and volumetric degradation, however it has been experimentally proven and it is well known that volumetric degradation is the main failure mode of a hydraulic pump, hence a novel concept for measuring volumetric pump wear is presented in study.

Pump Health Monitoring

Most of the failure cases in the IGP, and in fact in all types of pumps, are reflected in the volumetric efficiency. Due to mechanical wear gaps between the different pump elements become larger, increasing the internal leakage of the pump and consequently decreasing the volumetric efficiency of the MPU. (Rundo & Corvaglia, 2016) present an overview of possible gaps in IGP.

Volumetric efficiency is defined as the quotient of effective flow rate $Q_{\text{effective}}$ and theoretical flow rate $Q_{\text{theoretical}}$

$$\eta_{\text{vol}} = \frac{Q_{\text{effective}}}{Q_{\text{theoretical}}} \quad (1)$$

The theoretical flow is calculated with the MPU-Speed n_{MPU} and the fixed known displacement of the pump V_{pump}

$$Q_{\text{theoretical}} = n_{\text{MPU}} \cdot V_{\text{pump}} \quad (2)$$

As shown in Figure 2 and with Eq. (2), the theoretical volumetric flow for the VSFD-EMP can be calculated because of the known pump speed and the fixed displacement. By continuously determining the volumetric efficiency, e.g. for predefined operating points (OP), the (volumetric) state of the pump can be compared to the previously defined volumetric efficiency limit $\eta_{\text{vol},\text{lim}}$, as seen in Figure 3.

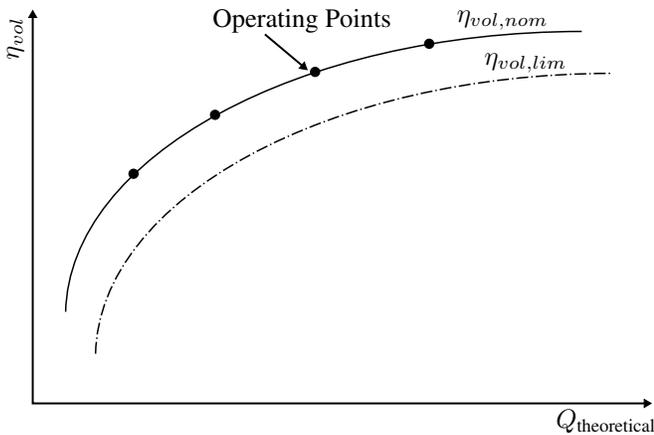


Figure 3. Volumetric efficiency

If the predefined limit is reached the MPU is replaced. Because it is assumed that the EMP is designed as a Line Replacement Unit (LRU), the specific reason for the deterioration in volumetric performance is not determined insitu. The MPU, as an LRU, will be disassembled and examined in more detail offboard in a second step. By checking similar OPs regularly even a prognostic about the pumps remaining remaining safe operating time ($\Delta\tau$), based on the distance between the computed volumetric degradation trend and the volumetric efficiency limit, can be computed. This is shown qualitatively in Figure 4.

The determined efficiency degradation trend of the pump is plotted over the flight cycles. A η_{vol} trend, in form of a (linear-) regression (LR) is then defined. By extrapolating the computed trend, the remaining operating life within safe limits $\Delta\tau$ is computed.

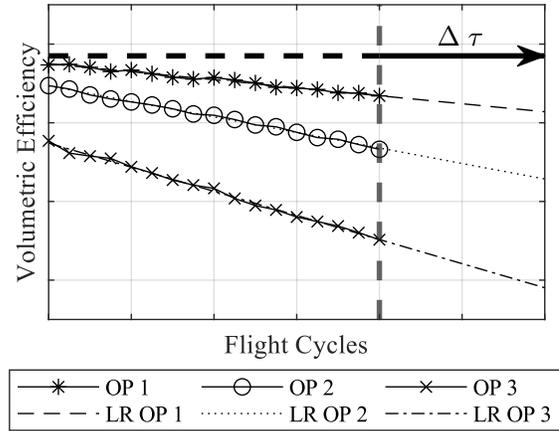


Figure 4. Volumetric efficiency trend for PHM

Furthermore, it should be noted that the theoretical flow can also be determined for an FSVD-EMP, however the position of the swashplate must be known. Consequently, for a FSVD-EMP, an additional sensor with very high accuracy would need to be installed to measure the position of the swashplate. This would increase the overall cost and complexity of the device. Therefore, the introduction of the VSFD-EMP enables a new approach to pump health monitoring.

In contrast to the determination of the theoretical volumetric flow, the determination of the effective volumetric flow is independent of the fixed displacement pump type.

3. MEASUREMENT OF THE EFFECTIVE VOLUMETRIC FLOW

As mentioned in the section before, the determination of the effective volumetric flow is critical for pump monitoring. This section gives a brief review of measuring methods. This is followed by an evaluation of the methods for use in aircraft hydraulics. Lastly, the most promising method is presented.

There are many ways to classify flow measurement devices. The classification shown in this study is presented in (Hardy, Hylton, McKnight, Remenyik, & R., 1999) and is based on the method used to extract the information from the fluid system. An overview of the classification and some examples of flowmeters can be seen in Figure 5.

Inferential flowmeters measure a physical quantity other than flow or velocity, whereupon the volumetric flow rate is then calculated. Energy-additive flowmeters transfer energy into the fluid. The effects of the flow on this energy are then used for the flow calculation. The volumetric flow is directly measured with Direct-measurement flowmeters.

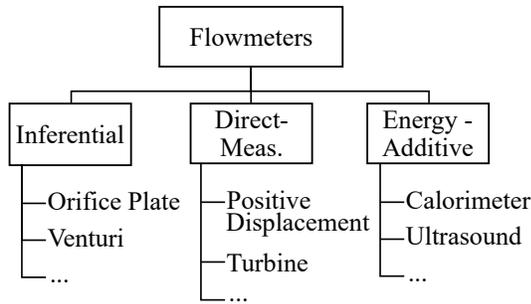


Figure 5. Flowmeter classification acc. (Hardy et al., 1999)

In situ Measurement

The selection of a volumetric flow sensor for aircraft use is more stringent than for common industrial applications. There are many possible considerations that must be taken into account, such as price, weight, complexity, and reliability. This study focuses mainly on two aspects. The first is that the chosen method should be approved for onboard use (In-situ Measurement). The second is the suitability of the measurement method within the hydraulic system. This means that under no circumstances should the chosen method interfere with the operation of the hydraulic system itself.

Ultrasonic-based flowmeters, have been successfully tested at the Institute of Aircraft Systems Engineering under laboratory conditions. They do not interfere with the operation of the hydraulic system as they are a non-invasive method of flow measurement, but they are not approved for onboard use. All in all, energy-additive flowmeters require complex electronics which are costly and reduce the overall reliability of the measurement method.

Direct-measure flow meters, such as gear flow meters, are not approved for onboard use and will affect hydraulic fluid operation. Taking a gear flow meter as an example, when the gears fail and can no longer turn, additional resistance is created in the hydraulic system. This results in massive pressure losses that ultimately lead to a system shutdown. To solve this problem, an additional bypass is required, which increases the weight and overall complexity of the system.

Inferential flow meters, such as orifice or venturi flow meters, use a pressure differential to calculate volumetric flow. For example, an orifice plate creates a pressure drop between the upstream and downstream of the orifice plate. By measuring the pressure drop, temperature, and knowing the properties of the fluid under operating conditions, the flow rate is then calculated. The differential pressure sensor (DPS) and temperature sensor have no failure effect on the hydraulic system, as they are not invasive and have no moving parts which can fail. Differential pressure sensors are already used in aerospace to monitor filters and for force fight compensation on Primary Flight Control Actuators for active-active control of surfaces such as the rudder for example (Lauckner & Baumbach,

2010; Spitzer, 2018). Temperature sensors have long been used in aircraft hydraulics. Therefore, inferential flowmeters are theoretically approved for onboard use but installing orifice plates in the hydraulic system is also non-practical because they cause unwanted pressure losses.

Therefore, the choice of a resistance to create a pressure differential and use of this principle to monitor pump condition must be made carefully.

Possible Resistances for Flow Estimation

As mentioned previously, pressure losses on the high pressure side of the system are not desirable. This is true for any aircraft hydraulic system, but the choice of resistance is system dependent. This study considers the scenario of a More Electric Aircraft (MEA) with a distributed eH-system architecture as introduced in (Trochelmann, 2020). A short description of the systems is given.

A center-zone system supplies hydraulic power to the main landing gear (MLG) and the power control unit (PCU), which are the consumers of the hydraulic system. Since the system is active only during short periods before takeoff (slats, flaps extension), after takeoff (MLG and slats/flaps retraction, before landing (MLG and slats/flaps extension), and after landing (slats/flaps retraction), a selector valve with a heating restrictor (Heating Valve) is installed to isolate the loads from the MPU, cf. Figure 6. The heating restrictor generates throttle losses and heats the fluid, e.g. before landing. The heating valve provides a resistance in the system that does not cause pressure losses when hydraulic fluid is directed to the consumers, thus enabling flow measurement. It has to be noted, that this principle can be basically used for every hydraulic system, as long as a valve decouples the consumers from the HePP.

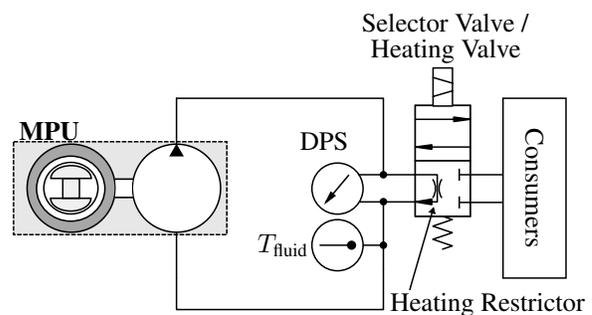


Figure 6. Simplified representation of a center-zone system with heating valve

The tail system consists of an eHEPP that supplies hydraulic fluid to the elevators and a rudder actuator. These consumers are always active and do not require a heating valve. There are no resistances in the hydraulic system that can be used to determine the effective volumetric flow rate. Therefore, as an alternative to the heating valve, a flow resistance is built

into the eHEPP itself. The flow resistance can be placed, for example, between the reservoir and the suction port of the pump. This is shown in Figure 2 as a generic flow resistance. The resistance creates the necessary pressure drop to be able to estimate the effective volumetric flow.

Comparison of Resistances

There are two main differences between the two possible presented resistances. The first difference is the method of calculating the flow rate. On the one hand, the heating valve has known characteristics, so the calculation is simple. The calculation can be done with the usual orifice equation

$$Q_{HV} = \alpha \cdot A \cdot \sqrt{\frac{2 \cdot \Delta p}{\rho(p, T_{fluid})}}, \quad (3)$$

where α is the flow coefficient, A is the orifice area, Δp is the pressure difference, and ρ is the density of the fluid. On the other hand, it is assumed that the flow resistance has a complex geometry, which makes the calculation of the volumetric flow cumbersome. The second difference is that the heating valve is part of the hydraulic system, while the generic flow resistance is part of the eHEPP. This makes the flow resistance system independent, which is a significant advantage over the heating valve. Therefore, the flow resistance within the eHEPP is chosen for flow estimation. In order to overcome the challenge regarding the computation of the volumetric flow, a different approach is taken, which will be discussed in the next section.

4. VOLUMETRIC FLOW ESTIMATION WITH THE FLOW RESISTANCE

The characterization of the flow resistance is one of the most relevant steps for the flow estimation. Because the characterization is performed experimentally, the test rig at the Institute of Aircraft Systems Engineering (FST) of the Hamburg University of Technology (TUHH) is introduced. This is followed by the actual characterization.

4.1. System-Test Rig

The system test rig is shown in Figure 7. It consists of two separate component test rigs. On the first test rig (eHEPP test rig), two parallel VSFD MPUs are installed. Each pump has a separate high pressure and suction line. This test rig also has other relevant power generation components such as reservoir, check valves, relief valves and filters. As described in the previous section, a servo valve for load emulation is also present in this test rig. The second test rig represents the hydraulic consumers of the tail section of the aircraft. It includes two elevators and one rudder of the aircraft. These are supplied with hydraulic fluid from the eHEPP test rig.

Each test stand has its own control unit. For data exchange and synchronization, the control models are connected via a CAN bus system. The data is then recorded centrally in the control unit of the eHEPP test rig.

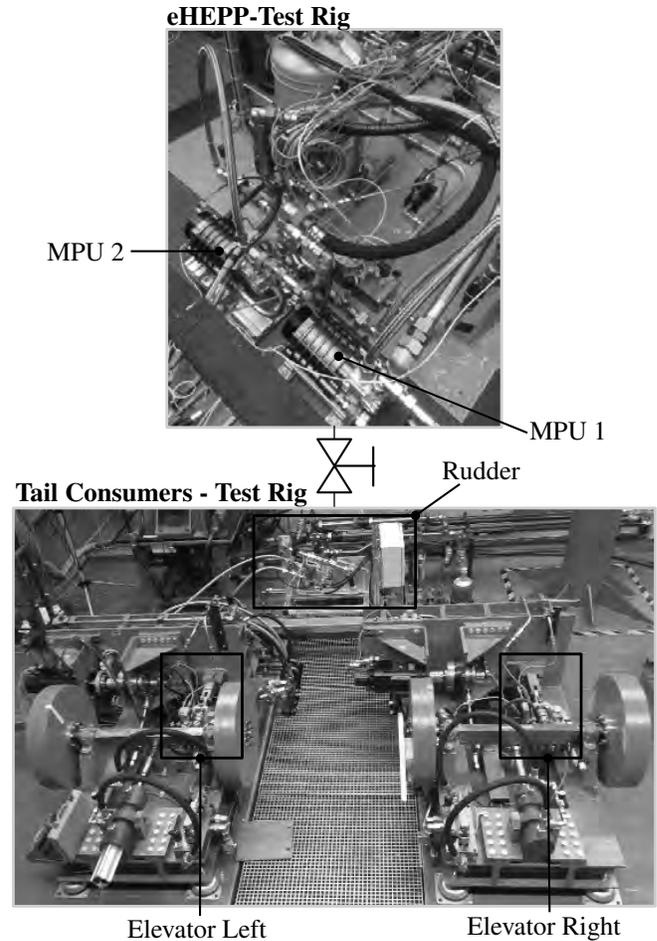


Figure 7. System-test rig at FST

4.2. Characteristic Map of the Flow Resistance

Similar to the heating valve, it can be assumed that the pressure losses in the flow resistance depend on the viscosity and density of the fluid. Both properties depend on the pressure and temperature of the fluid itself. Therefore, a flow calculation with a known analytical solution is not straightforward. The chosen solution to this problem is to measure the characteristic map of the flow resistance. This approach implicitly takes into account the properties of the fluid and their effect on the pressure difference.

The determination of the characteristic map is carried out with the eHEPP test rig at the FST with a MPU. Several operating points of the EMP are set with the load servo valve.

The aim is to achieve a homogeneous distribution of the calibration points in form of

$$Q_{\text{effective}} = f(\Delta p, T_{\text{fluid}}). \quad (4)$$

The differential pressure Δp , fluid temperature T_{fluid} and volumetric flow rate $Q_{\text{effective}}$ are measured over a period of approximately five seconds. Table 1 shows the properties of the used sensors.

Table 1. Accuracy of sensors

Measurement	Sensor	Accuracy
Differential Pressure Sensor	Δp	$\pm 0.2\%$ FS
Temperature Sensor	T_{fluid}	0.5°C
Volumetric Flow Sensor	$Q_{\text{effective}}$	0.3% of meas. value

It should be noted that the volumetric flow is measured with a gear flowmeter mounted on the high pressure side of the system. For each operating point, the average value of pressure, temperature and volumetric flow is calculated. This results in a characteristic map of the chosen flow resistance, as shown in Figure 8.

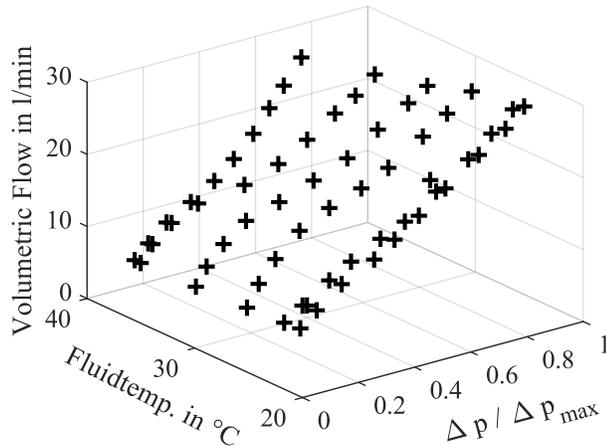


Figure 8. Calibration data

The temperature dependence of the pressure losses can be clearly seen. Especially at higher flow rates, the pressure drop is lower at higher temperatures. This is to be expected due to the lower viscosity and density of SKYDROL at higher fluid temperatures.

An initial assessment of the chosen method for determining the effective volumetric flow is the total measurement uncertainty for each measurement $\sigma_{total,x}$, where x stands for the differential pressure measurement, the fluid temperature or the volumetric flow. This is composed of the standard deviation of the mean $\sigma_{\bar{m}_w}$ and the sensor uncertainty σ_S

$$\sigma_{total,x} = \sqrt{\sigma_{\bar{m}_w}^2 \cdot \sigma_S^2}. \quad (5)$$

Low total uncertainty values are achieved for all calibration points and all measurements. Therefore, the calibration points can be used for the calculation of the volume flow. However, using individual measurements as a map to determine the flow rate can lead to inaccuracies (e.g., interpolation and extrapolation). A suitable method to approximate the points, reduce the inaccuracy and avoid large computational capacities is still necessary. This would also enable the online implementation of the monitoring concept.

4.3. Approximation of the Characteristic Map

There are several methods to approximate the calibration data for online flow estimation. The various approximation methods are examined using the *Matlab Curve Fitting Toolbox*. Many options are available to evaluate the goodness of fit, e.g. statistics, residual analysis, confidence and prediction bounds. On the one hand, statistical analysis and confidence bounds are numerical methods for determining the goodness of fit. On the other hand, residual analysis and prediction bounds are graphical methods. Depending on the data and the fitting requirements, a suitable method of approximation and evaluation is chosen.

In this case, a simple, easy-to-understand model is desired. Since it does not need to have physical meaning, a polynomial approximation is chosen. For the evaluation, a simple residual analysis is performed. The reason for this selection is that this method makes it simple to evaluate the effects of the approximation on the monitor. For example, a deviation between calibration data and approximation at low volumetric flows leads to a large relative error. The same deviation leads to small errors at high volumetric flows. A residual analysis is therefore used for an initial evaluation of the chosen polynomial. Figure 9 shows the calibration data with the selected area fits.

As can be seen in the figure, the calibration data were approximated using two surface fits. The reason for this is that the approximation of all calibration points with one surface (one fit) did not meet the main requirement, particularly large errors were obtained for small volumetric flow rates. A better overall approximation is achieved with two separate approximations. For this purpose, the calibration data is divided into two sets. The first set contains all calibration points with a normalized pressure drop of approx. 0.25 or less. The second set contains all other values. It has been shown that a bivariate polynomial with a total degree of three is a good approximation for the available low and high volume flow data. The two variables are the pressure difference Δp and the outlet temperature T_{fluid} . However, the coefficients of the polynomials differ.

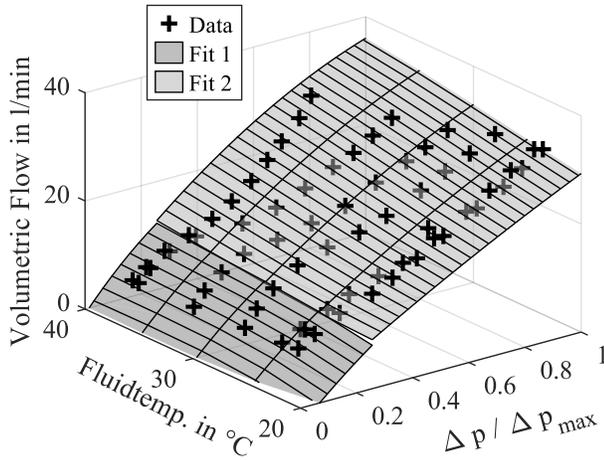


Figure 9. Characteristic map of the flow resistance

The residual analysis for all fluid temperatures is shown in Figure 10.

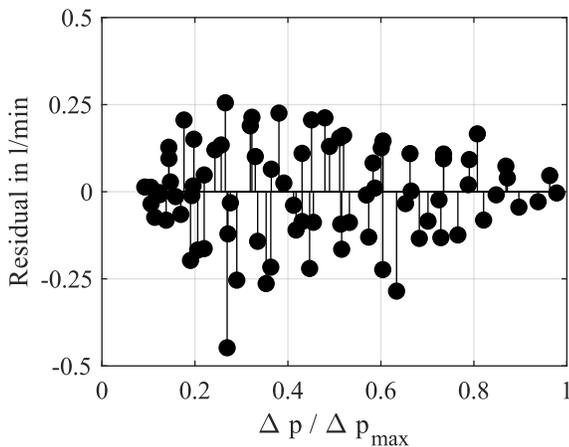


Figure 10. Absolute error calibration data-polynomial

For low pressure drops, the maximum residual value is less than $0.25 \frac{l}{min}$, resulting in a relative error of less than 5%. For higher flow rates and higher pressure drops, a similar residual value can be found, resulting in a much smaller calculated error approx. 1%. The created characteristic map of the flow resistance is then implemented.

5. ONLINE FLOW ESTIMATION

The approximation chosen in Section 4 is implemented in the control model of the eHEPP test rig, allowing the online computation of the volumetric efficiency of the MPU 2. The results of a test campaign with the real consumers are shown in this section.

5.1. Pump PHM with Real Consumers

In a system with real consumers, the operating points with low dynamics, which is expected to deliver a more accurate estimation of the volumetric flow, depend on the system. For example, in the blue hydraulic system of the A320, the extension of the slats before takeoff and the compensation of leakage during cruise are suitable operating points for volumetric flow estimation. In contrast, for the tail system, only the compensation of leakage during cruise is suitable. To be able to set other operating points, a targeted procedure of the actuators is an alternative. This could be implemented as an automatic pre-flight test. Now it has to be checked up to which actuator speeds, suitable operating points for volume flow estimation are reached (quasi-stationary states). This is checked on the FST System test rig.

Table 2 shows the selected profiles of the actuators for the investigation of estimation based on the pressure drop across the selected flow resistance. The operating points cover different load flow rates by varying both the combination of actuators and the rate of change. This can be changed depending on the OP to be monitored.

Table 2. Possible operating points for pump condition monitoring

Time	OP 1	OP 2	OP 3	OP 4	OP 5
$x_{cmd,elevators}$ in mm	20	-40	20	-40	20
$v_{elevators}$ in mm/s	15	30	40	40	40
$x_{cmd,rudder}$ in mm	11	11	11	55	-35
v_{rudder} in mm/s	0	0	0	30	88

The profiles are selected so that the movement of the consumers (elevators and rudder) ends simultaneously. This reduces dynamic effects and achieves homogeneous curves. OP 5 corresponds to the operating point where the maximum volumetric flow due to movement of the actuators is achieved.

For the evaluation of the flow estimation, the measurement data is first filtered with a low pass filter

$$H(s) = \frac{1}{1 + \frac{s}{\omega_c}} \tag{6}$$

where ω_c represents the cutoff frequency. The cut off frequency is chosen so that measurement noise is suppressed but the dynamics of the system are not. The characteristic map is implemented in the test rig control model. Similar to Figure 2 the differential pressure, fluid temperature and pump speed is used to compute the volumetric efficiency.

The results of the investigation are shown in Figure 11. The figure shows the position of the actuators, the online volumetric flow estimation and the calculated volumetric efficiency. For the volumetric flow rate and volumetric efficiency, the estimate is compared with the gear flow meter.

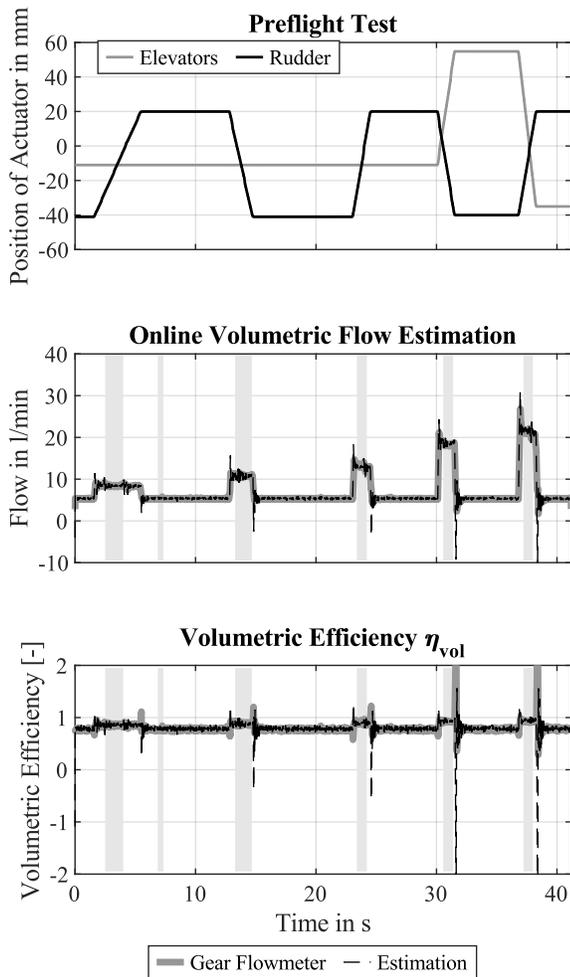


Figure 11. Comparison flow estimation vs. flowmeter with real consumers

The chosen approximation with the two polynomials shows the desired behavior. For small and large volume flows, the approximation achieves high accuracy. This is particularly evident for quasi-stationary and less dynamic operating points and are marked by a light gray background.

Large differences between the estimated and the measured flow are observed when much higher dynamics are present. The main reason for this is the different installation position of the DPS and the gear flow meter. While the DPS is located in the suction line, the gear flow meter is installed on the high pressure side of the system. The effects of the inertia of the fluid becomes particularly apparent when the pump speed is drastically reduced (the actuators reach the desired set position). Because negative pressure differentials are achieved, the results are to negative flow rate estimates. A similar effect also occurs when the pump is accelerated (the actuators start to move). In this case, high pressure differences are measured

and higher volume flow rates are estimated. Nevertheless, this is not an indication of an erroneous estimate. The errors that occur in highly dynamic processes are caused by the position of the DPS and the working principle. To be noted here is, that the tracking of volumetric pump efficiencies is not required in all states including transients. The tracking is essential in the quasi stationary main operating points of the hydraulic system, which indicates the pump wear sufficiently.

Lastly, similar to the section 2.2, the volumetric efficiency for the different OPs are calculated. For this purpose, the mean value of the volumetric efficiency during (mostly) quasistationary phases is computed. The result are shown in Figure 12.

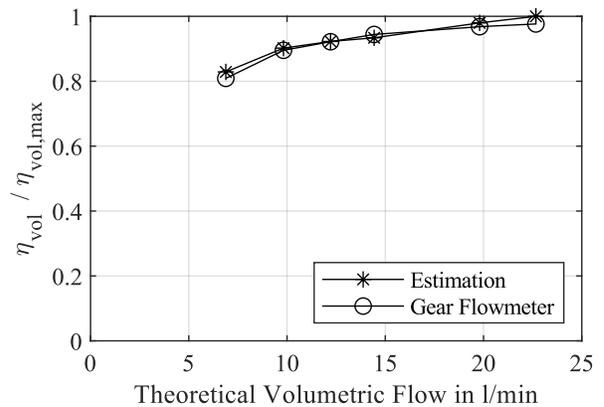


Figure 12. Comparison of pump efficiency: estimation vs. gear flowmeter

The figure shows the results for the gear flowmeter and for the estimation. As it can be seen the characteristic curves match well. This means that the chosen method for the estimation of the volumetric flow can be used for the monitoring of the condition of the hydraulic pump in aircraft hydraulics.

6. LESSONS LEARNED

In this section some of the lessons learned during this study are summarized.

Integration of DPS: The sensitivity of the integration of the DPS was also investigated in this study because there are two MPUs in the EMP test rig. Some small changes in the integration of the DPS between two MPUs lead to large differences in the estimated volumetric flows. Therefore, an optimal integration of the DPS is strongly recommended.

Inaccuracy for low volumetric flows: Depending on the objective for determining the efficiency (low or high volume flows), the quality of the approximation must be taken into account. As already described, an approximation with high residuals leads to significantly higher errors at low volumetric flows.

Measurement of Actuator Leakage: The proposed method allows not only to determine the volumetric efficiency of the pump, but also to determine the internal leakage of the Control Servo Actuators (CSA). Depending on the mode (active or inactive) of the actuators, the internal leakage of all active actuators can be determined. Even if the CSA with higher leakage cannot be isolated, the process of isolation can be accelerated.

7. CONCLUSION

In this paper, a new concept for pump health monitoring is proposed. The concept is based on the insitu estimation of the effective volumetric flow. The estimation leads to the calculation of the volumetric efficiency, which is then used to determine the volumetric pump efficiency with the known speed and displacement. In the first section, various options for determining the effective volumetric flow rate were analyzed with respect to their application in an aircraft hydraulic system. It was found that none of the conventional methods can be used in aircraft hydraulics, but the rugged and well mature principle with high reliability of measuring differential pressure and fluid temperature in aerospace may be employed. The main challenge in this instance is to find the appropriate resistance. In this case, a generic flow resistance was used. The main advantage of such a resistance is the system independence. Depending on the resistance, the corresponding pressure drop and desired accuracy, the choice of a suitable pressure sensor is very important.

In the second part, the implementation of online flow calculation is presented. For the estimation, the characteristic map of the resistance is determined. This is performed experimentally. The data is used as calibration points for the characteristic map, which is approximated using two higher order polynomials. The approximation is the basis for the online implementation. It has been shown that the quality of the approximation significantly affects the accuracy of the estimate. High accuracy is especially required for small volume flows. Finally, the online estimation of volumetric efficiency was tested with real aircraft consumers. The estimation shows high accuracy in an ideal test (load servo valve) and with real consumers.

The tests performed so far were done with an MPU that shows no degradation. Although the initial results are very promising, the TUHH Institute of Aircraft Systems Engineering intends to test the monitor with a custom-built test rig that emulates pump degradation. In addition, the limits for the volumetric efficiency as well as the influence of the fluid temperature will be determined. The presented PHM method uses an additional sensor (DPS) to determine the condition of the pump and since there are no available options yet known

without an additional sensor, alternative solutions for this method will be investigated.

ACKNOWLEDGMENT

This work was funded by the German Federal Ministry of Economic Affairs and Energy (BMWi) within the MODULAR project (contract code: 20Y1910G) in the national LuFo VI-1 program. Their support is greatly appreciated. Thanks also go to Nils Trochelmann, who played a vital part in the measurement campaign.

Supported by:



REFERENCES

- Hardy, J. E., Hylton, J., McKnight, T. E., Remenyik, C. J., & R., R. F. (Eds.). (1999). *Flow measurement methods and applications*. Wiley.
- Lauckner, S., & Baumbach, V. (2010). Recent advances in commercial aircraft hydraulic systems. In *7th international fluid power conference*.
- Poole, K. (Ed.). (2015). *Modellbasierte entwicklung eines systems zur zustandsdiagnose und -vorhersage für die hydraulische energieverorgung in verkehrsflugzeugen*. PhD thesis TUHH.
- Rundo, M., & Corvaglia, A. (2016). Lumped parameters model of a crescent pump. *Energies*(11).
- Spitzer, C. R. (Ed.). (2018). *Avionics: Elements, software and functions*. CRC Press.
- Trochelmann, N. (2020). Thermal-dynamic investigation of advanced system control strategies for decentralized electro-hydraulic power generation in more electric aircraft. In *Proceedings of the aerospace europe conference 2020*.
- Trochelmann, N., Bischof, P., Thielecke, F., Metzler, D., & Bassett, S. (2018). A robust pressure controller for a variable speed ac motor pump – application to aircraft hydraulic power packages. In *Bath/asme 2018 symposium on fluid power and motion control (fpmc)*.
- Trochelmann, N., Rave, T., Thielecke, F., & Metzler, D. (2017). An investigation of electro-hydraulic high efficient power package configurations for a more electric aircraft system architecture. In *Deutscher luft- und raumfahrtkongress*.

Hybrid Fault Prognostics for Nuclear Applications: Addressing Rotating Plant Model Uncertainty

J Blair¹, B Stephen², B Brown³, A Forbes⁴, and S McArthur⁵

^{1,2,3,5} *University of Strathclyde, Glasgow, United Kingdom*

j.blair@strath.ac.uk

bruce.stephen@strath.ac.uk

blair.brown@strath.ac.uk

s.mcarthur@strath.ac.uk

^{1,4} *National Physical Laboratory, Teddington, United Kingdom*

alister.forbes@npl.co.uk

ABSTRACT

Nuclear plant operators are required to understand the uncertainties associated with the deployment of prognostics tools in order to justify their inclusion in operational decision-making processes and satisfy regulatory requirements. Operational uncertainty can cause underlying prognostics models to underperform on assets that are subject to evolving impacts of age, manufacturing tolerances, operating conditions, and operating environment effects, of which may be captured through a condition monitoring (CM) system that itself may be degraded. Sources of uncertainty in the data acquisition pipeline can impact the health of CM data used to estimate the remaining useful life (RUL) of assets. These uncertainties can disguise or misrepresent developing faults, where (for example) the fault identification is not achieved until it has progressed to an unmanageable state. This leaves little flexibility for the operator's maintenance decisions and generally undermines model confidence.

One method to quantify and account for operational uncertainty is calibrated hybrid models, employing physics, knowledge or data driven methods to improve model accuracy and robustness. Hybrid models allow known physical relations to offset full reliance on potentially untrustworthy data, whilst reducing the need for an abundance of representative historical data to reliably identify the monitored asset's underlying behavioural trends. Calibration of the model then ensures the model is updated and representative of the real monitored asset by accounting for differences between the physics or knowledge model and CM data.

Jennifer Blair et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

In this paper, an open-source bearing knowledge informed machine learning (ML) model and CM datasets are utilized in an illustrative bearing prognostic application. The uncertainty incurred by the decisions made at key stages in the development of the model's data acquisition and processing pipeline are assessed and demonstrated by the resultant impact on RUL prediction performance. It was shown that design decisions could result in multiple valid pipeline designs which generated different predicted RUL trajectories, increasing the uncertainty in the model output.

Index Terms—bearing prognostics, condition monitoring, hybrid systems, model calibration, uncertainty capture.

1. INTRODUCTION

Most often, asset maintenance is conducted reactively, whereby corrective maintenance is conducted once a failure has occurred (Canada Nuclear Safety Commission, 2012). In a nuclear power plant (NPP), an unexpected outage of an asset can be expensive due to lost revenue from interrupted generation with downtimes being potentially lengthened by the requirement to: retrospectively identify the root of the fault, source required components and perform the maintenance action. With many NPP's coming to the end of their designed lifetime, many operators are utilising CM and condition based maintenance (CBM) techniques to justify and manage NPP lifetime extensions and to avoid unplanned outages (Coble, Ramuhalli, Bond, Hines, & Upadhyaya, 2015). This requires aging assets to be closely monitored to estimate asset health and ensure extension plans are affordable.

A common asset in NPP's are rotating plant (e.g. motors, turbines, centrifugal pumps, fans), which are prone to bearing failure (Yung & Bonnett, 2004). These could be turbine or motor driven pumps which form part of a larger gen-

eration or cooling system. Despite being relatively simple components, bearings are largely responsible for the reliable operation of rotating plant by supporting huge loads to reduce friction on downstream components (Jammu & Kankar, 2011). As such, if bearing faults are left untreated, damage could propagate through the drivetrain and create wider system complications in more expensive components, such as the gear box (Rexnord Industries, LLC, Gear Group, n.d.). Cascading failures would lead to expensive and lengthy maintenance intervention which would cause disruption to plant generation and incur additional regulatory reporting overhead.

CBM and data based analytics can be used to estimate the RUL of rotating plant bearings which, if effective, can provide sufficient warning of an impending failure and an indication of the type of failure developing. An operator can incorporate this into their resource scheduling and budgeting actions to ensure the asset is taken offline and serviced while minimising disruption to NPP operation. However, developing and applying this approach requires access to data, and with specific regards to NPP's, these systems were designed before modern digital sensing and monitoring techniques were available for the hostile environments where they operate, which is an additional consideration that can impact upon the associated data acquisition components. This can result in operators making decisions on unhealthy data collected from NPP's which are not ideally designed for modern sensing systems, adding additional uncertainty to maintenance plans.

Sources of uncertainty can impact the data acquisition pipeline at every stage, including: the choice of sensor type and placement; the chosen sampling rate; data pre-processing steps to present the data in a specific format; and, the metric(s) used by the analytics to convey information to an operator. Design choices at each of these stages offer a trade off, which will incur uncertainty in the output of the pipeline at each stage and can be compounded by the interaction between upstream and downstream pipeline stages. In addition to this, data based analytics generally do not attribute a measure of confidence in their output, making it difficult to determine if the analytics are performing poorly in a sub-optimal pipeline. This makes ML outputs difficult to trust for inclusion in risk and cost assessments. Also they do not provide the operator with relevant information that could allow future improvements to the pipeline to be made.

2. CONTRIBUTION

The contribution of this work is not the creation of a novel RUL technique, but to demonstrate and quantify the confidence associated with the application of existing hybrid RUL approaches with the associated data acquisition pipeline decisions. Confidence can be undermined by these choices, which impact the performance of the underpinning model and can

reduce the operators trust in the whole decision support system. Without sufficient trust, especially in the heavily regulated nuclear engineering environment, decision support tools will not be utilised to support maintenance scheduling activities. As such, the methodology presented in this paper is concerned with investigating the uncertainty in analytic design and deployment by capturing the sources of uncertainty and demonstrating how these impact on an uncertainty budget for the whole data to decision pipeline rather than just the output of the ML model. In this work, the uncertainty in the model performance due to the whole pipeline design is captured by analysing the quantiles of the model outputs under different data acquisition pipeline designs. Evidence is presented in the form of case-studies using open-source, curated test rig data to reduce the impact of excessive operational noise, that were performed to evaluate data pipeline uncertainty.

3. LITERATURE

The literature review covers research trends in bearing prognostics applications, with a focus on data-based methods as these require access to healthy CM data. This is supported by a section on hybrid modelling where knowledge- and data-based methods are combined, and how diverse approaches may be combined for prognostic applications. Finally, uncertainty capture methods with particular focus on computer modelling prognostic methods is presented.

3.1. Data-Based Bearing Prognostics

Bearings are subject to high stress operating conditions which makes failures common. These can manifest due to overloading or imbalanced loading, lubrication issues due to insufficient lubrication, contamination or sealing failures. Bearings are mechanical faults and mechanical failures are most commonly monitored via vibration monitoring, although have been approached using temperature, oil analysis and acoustic emission approaches (Kumar et al., 2019). Vibration monitoring, while subjected to the robustness and cost of the sensor system, allows changes in bearing health to be observed immediately and has been proven as a reliable method for bearing fault prognosis. Temperature based schemes are most useful for end of life where the fault has progressed significantly, oil analysis methods require the bearings to have a dedicated supply system and acoustic emission requires access to high quality measurements (Jammu & Kankar, 2011).

A survey of 274 prognostic approaches by (Lei et al., 2018) separated works into statistical-, AI-, physics- and hybrid-based approaches, with 56% contribution from statistical based methods, and 26% from AI based approaches which both rely heavily on available CM data. ML or Deep Learning (DL) approaches are gaining increasing popularity as they can handle complex prognosis problems which may be traditionally difficult to create reliable physics or statistical models for, how-

ever due to their black-box nature it is difficult to justify their usage in safety critical applications. The approaches which gained the most attention for machine prognosis in (Lei et al., 2018) review were Artificial Neural Networks, Neuro-Fuzzy systems (both DL methods), Support Vector Machines (SVM), K-nearest neighbour (k NN) and Gaussian Process Regression. DL approaches require access to large quantities of high quality, representative data which can be unobtainable in some industrial settings, however can produce excellent RUL predictions in return. ML models such as the SVM and k NN methods can provide better performance in cases with limited access to representative data, however are subject to appropriate kernel and parameter selection (Nisbet, Elder, & Miner, 2009). Gaussian Process Regression are computationally expensive when utilising large number of samples due to a required matrix inversion, but is a flexible method that can be updated with new data, adapt to limited data and incorporate uncertainties (Hart, 2018).

3.2. Hybrid Models

A single knowledge-, physics- or data- based approach is unlikely to provide effective system coverage for multiple failure modes and fault types. Utilising a combination of approaches aims to leverage the relative advantages of each individual method while limiting the impact of their respective weaknesses (Baur, Albertelli, & Monno, 2020). (Goebel, Eklund, & Bonanni, 2006) found that combining a bearing physics of failure model with an empirical method based on measured data (Dempster-Shafer Regression) produced more accurate RUL prediction results than either method independently.

The method of combining two or more of these methods in a hybrid approach varies and tends to be application specific due to the relatively early development stage of the research field as shown by the small (8 %) contribution to the canvassed literature in (Lei et al., 2018). As such, many methods of creating hybrid models are being explored, such as utilising one model to estimate the asset health state and another for RUL estimation; combining the RUL estimates from multiple methods; or utilising one method for short-term forecasting and another method for long-term forecasting (Ramuhalli, Walker, Agarwal, & Lybeck, 2020). Of particular interest in this work is the combination of knowledge- and data-based approaches. Incorporating domain knowledge into data-driven approaches allows known trends and rules that govern the degradation patterns to be encoded to support the prognostic tool in identifying and predicting the failure dynamics of well understood failure modes. The data-driven component can provide the needed flexibility to apply and extrapolate these rules into an RUL estimate tailored to the monitored asset, while providing capability to identify new failure modes not included in the encoded expert knowledge (Liao & Köttig, 2014).

3.3. Uncertainty Capture in computer models

ML models tend to produce point estimates which does not provide information about the likely distribution of potential predictions. Attributing confidence intervals to output predictions (typically corresponding to a confidence level of 95 % (JCGM Working Group 1, 2008)) provides more appropriate information about the anticipated range of outputs and expected value of the prediction, allowing operators more agency to utilise the results. Bayesian methods are usually incorporated into prognostic approaches to handle uncertainty capture and propagation, such as in Bayesian Networks, Bayesian Neural Networks and Kalman/particle filtering algorithms. Non-Bayesian methods that have been used include Monte Carlo based, bootstrapping or closed-form mathematical solutions. A major drawback of these approaches are the lengthy and difficult process of collecting and formalising prior knowledge and assumptions which may be impractical for some complex system applications (Zhao et al., 2021).

Additionally, computer models themselves introduce sources of uncertainty. This was demonstrated by (Kennedy & O'Hagan, 2001) who utilised a Bayesian approach to computer model calibration that incorporated all forms of uncertainty previously discussed in the research space. These included: parameter uncertainty, where the value of context specific features are unknown but assumed; random effects, where the real process may experience random fluctuations given the same experienced conditions; model inaccuracy, where the complexity of the model is unable to truly reflect the real process; data collection errors, where there are sources of uncertainty in the CM data; and uncertainty of the unseen code output, where there exists unprocessed and potentially more optimal configurations of the model.

4. UNCERTAINTY IN HYBRID MODEL DATA PIPELINES

The proposed methodology to assess the uncertainty is presented in the form of a case study that investigates the impact of decisions made in the data acquisition and processing pipeline through the resulting uncertainty in the RUL prediction for motor bearing prognostics.

4.1. Condition Monitoring Datasets

Two open source bearing prognostics datasets are used in this work: NASA IMS (Lee, J. , Qiu, H. , Lin, J. and Rexnord Technical Services, 2007) and NASA FEMTO (NASA Ames Prognostics Data Repository, 2012). Both datasets observe run to failure experiments for bearings with no initial defects. Each data set has visibility of the bearings failures by vertically and horizontally mounted accelerometers (termed 'x-axis' and 'y-axis' respectively), with limited access to the vertical data for the IMS dataset. Four distinct bearing failures are observed in the IMS dataset, with two occurring concurrently, while the FEMTO dataset contains 17 run to failure

examples. The IMS failures were accelerated due to intensive, but in specification, bearing loading conditions, while the FEMTO dataset was created using the PRONOSTIA test rig which artificially overloaded the bearings to further accelerate wear.

4.2. Existing Hybrid Model

4.2.1. Combining Knowledge- and Data-driven Components

An open source hybrid RUL model consisting of a novel Weibull-based loss function for Neural Networks (NN) by (Hahn & Mechefske, 2022) was chosen as the basis for this study. Utilising a Weibull distribution to capture domain knowledge from the field of reliability engineering, the authors create 9 NN loss functions to evaluate the success of their knowledge informed ML model for bearing prognosis on the IMS and FEMTO datasets. The knowledge component of the hybrid model is captured by using the data to calibrate the Weibayes equation (Abernethy, 2004) shown in equation 1. The one parameter Weibayes has been shown to produce accurate results for a small number of failures (<20) where the estimated value of shape parameter, β , is representative of the true system behaviour (Abernethy, 2004). The value of β was fixed at a value of 2 in (Hahn & Mechefske, 2022) due to model stability concerns, and this value being deemed a reasonable shape estimate for ball bearing failures (Abernethy, 2004). The values of η and β are used to calculate the Weibull cumulative distribution function (CDF) in equation 2. The 9 loss functions are shown in figure 1 and are incorporated into the model as the loss function to be minimised by the NN in the back-propagation step.

$$\eta = \left[\sum_{i=1}^N \frac{t_i^\beta}{r} \right]^{\frac{1}{\beta}} \quad (1)$$

$$F(t) = 1 - e^{-\left(\frac{t}{\eta}\right)^\beta} \quad (2)$$

Where

- t = time or cycles,
- r = number of failed units,
- N = total number of failures plus currently running units (incomplete failures)
- η = maximum likelihood estimate of the unit characteristic life (63.2 distribution percentile)
- β = Weibull shape parameter, and
- F(t) is the Weibull CDF

4.2.2. RUL Estimation Procedure

(Hahn & Mechefske, 2022) conducted the following process to generate RUL estimates for both the IMS and FEMTO

Loss Function	Equation
MSE Loss (\mathcal{L}_{MSE})	$\frac{1}{n} \sum_{i=1}^n (t_i - \hat{t}_i)^2$
RMSE Loss ($\mathcal{L}_{\text{RMSE}}$)	$\sqrt{\frac{1}{n} \sum_{i=1}^n (t_i - \hat{t}_i)^2}$
RMSLE Loss ($\mathcal{L}_{\text{RMSLE}}$)	$\sqrt{\frac{1}{n} \sum_{i=1}^n (\log(t_i + 1) - \log(\hat{t}_i + 1))^2}$
Weibull Only MSE Loss ($\mathcal{L}_{\text{Weibull-MSE}}$)	$\lambda \frac{1}{n} \sum_{i=1}^n (F(t_i) - F(\hat{t}_i))^2$
Weibull Only RMSE Loss ($\mathcal{L}_{\text{Weibull-RMSE}}$)	$\lambda \sqrt{\frac{1}{n} \sum_{i=1}^n (F(t_i) - F(\hat{t}_i))^2}$
Weibull Only RMSLE Loss ($\mathcal{L}_{\text{Weibull-RMSLE}}$)	$\lambda \sqrt{\frac{1}{n} \sum_{i=1}^n (\log(F(t_i) + 1) - \log(F(\hat{t}_i) + 1))^2}$
Weibull-MSE Combined Loss ($\mathcal{L}_{\text{Weibull-MSE-Comb}}$)	$\mathcal{L}_{\text{MSE}} + \lambda \mathcal{L}_{\text{Weibull-MSE}}$
Weibull-RMSE Loss ($\mathcal{L}_{\text{Weibull-RMSE-Comb}}$)	$\mathcal{L}_{\text{RMSE}} + \lambda \mathcal{L}_{\text{Weibull-RMSE}}$
Weibull-RMSLE Loss ($\mathcal{L}_{\text{Weibull-RMSLE-Comb}}$)	$\mathcal{L}_{\text{RMSLE}} + \lambda \mathcal{L}_{\text{Weibull-RMSLE}}$

Figure 1. Loss functions from (Hahn & Mechefske, 2022)

Dataset	Train.	Val.	Test.
IMS	Run 2 (B 1) Run 3 (B 3)	Run 1 (B 3)	Run 1 (B 4)
FEMTO	Bearing1_1 Bearing2_1 Bearing3_1	Bearing1_2 Bearing2_2 Bearing3_2	Bearing1_3 Bearing2_3 Bearing3_3

Table 1. Data split between training, validation and testing

datasets. First, the input data from the horizontal sensors was processed into spectrograms to obtain the frequency representation of the vibration data. The number of input features was reduced by 'binning' the spectrogram into 20 bins, where the maximum value of the frequency bands included in each bin is taken as the value for that bin, repeated for each timestep. The response variable was the lifetime percentile status of the bearing, with 0 % being healthy bearing at the start of the experiment, to 100 % signifying the failure of the bearing at the end of the experiment. The training, validation and testing split of the datasets are shown in table 1.

The Weibayes equation was calibrated with the training data to be incorporated into the loss functions. To initialise and optimise the NN architecture, a random search was conducted to select from the hyper parameters shown in table 2 for each of the loss functions, which the authors set to 1000 in their study. The coefficient of determination (R^2) and Root Mean Squared Error (RMSE) were used to discard models that performed poorly, with models with a $R^2 > 0.2$ and $RMSE < 0.35$ progressed to the testing stage. After testing, the models were filtered again by the R^2 and $RMSE$ bounds before selecting a subset of the top performing models based on the R^2 metric. The authors found that the top performing loss function for the IMS dataset was the Weibull-RMSLE combined, and the Weibull-MSE combined for the FEMTO dataset, both containing the knowledge informed loss function.

Parameter	Selection Choice
Batch size	32, 64, 128, 256, 512
Learning rate	0.1, 0.01, 0.001, 0.0001
Lambda	Floating point number 0-3
Number of layers	Integer between 2 and 7
Number of units per layer	16, 32, 64, 128, 256
Probability of dropout	0.1, 0.2, 0.25, 0.4, 0.5, 0.6

Table 2. NN Architecture Hyperparameter Options Table from (Hahn & Mechefske, 2022)

4.3. Pipeline Design Uncertainty

For this sensitivity study, the data acquisition pipeline design was varied, considering the following stages and settings, also summarised in table 3.

4.3.1. Dataset

The FEMTO and IMS datasets were chosen due to initial bearing states with no faults; their curated, open source nature; but also their differences in aging methods, timescales and number of recorded failures. In the IMS dataset, the bearings are operated under their maximum specified operating condition limits and failed after their design lifetime (in number of revolutions). This represents scenarios where the bearings are operated in an unhealthy but within technical specification manner. However, as it took weeks to months to observe these failures, only 4 failures over 3 runs were observed, severely limiting the analytic’s scope to learn from a diverse sample of run-to-failure trajectories. This issue is reversed for the FEMTO dataset, where 17 distinct failures were observed due to the run-to-failure process taking several hours. However, the conditions the bearings were operated in would not be practical in an industrial setting. Data pipeline choices at this stage investigate the impact on the analytics RUL performance due to the amount and nature of the failures observed, and how the analytics perform on the different methods of accelerated lifetime testing.

4.3.2. Sensor Channel

Both datasets have access to vertically and horizontally aligned vibration sensors (noting limited availability for the IMS dataset). Depending on the nature of the fault, ML models may be more successful in identifying failure signatures in one axis over another, leading to more reliable RUL estimates if measurement data is available for this orientation. However, it is not always feasible or maintainable to retrofit assets with extensive sensor coverage, meaning the developing failure may not be measured from the most suitable angle. With no prior knowledge of the bearing failure, data pipeline choices at this stage investigate the consequences on the RUL estimate of having limited, and potentially inadequate, sensor coverage of an impending failure.

4.3.3. Data Sampling

In an ideal scenario, condition monitoring would consist of high resolution, continuous measurement to ensure that as much data is available to the prognostic algorithms as possible. In practice, this would generate enormous volumes of data that would be impractical to transmit, process and store, while potentially providing diminishing returns on the useful information contained in the data streams. Communications and storage infrastructure is limited in an industrial setting where fleets of assets are expected to be monitored simultaneously. At this stage of data pipeline uncertainty assessment, comparisons are made for RUL estimates where 1/8, 1/4, 1/2 and no data is lost due to these constraints.

4.3.4. Spectrogram Bin Count

The spectrogram binning process from (Hahn & Mechefske, 2022) allows the frequency domain information from the full spectrogram to be used while condensing this information into a more manageable number of input features to the ML stage. This forms a trade off between the amount of information lost in the binning process, and the dimensionality. The spectrogram bin count is chosen to be 10, 20 (as original author) and 40, to compare how the RUL is impacted by this trade off.

4.3.5. Hyperparameter Optimisation

NNs are computationally expensive to train, and it may be infeasible to evaluate a large selection of models in order to optimise the selected hyperparameters. Selecting a sub-optimal model will impact the quality of the RUL estimate. The original author runs a parameter search by selecting n combinations of model hyperparameters (table 2), then filtering out models with unsatisfactory performance. Computational limitations may make training many models to allow the most optimal hyperparameters to be chosen an unfeasible action to take. This stage of the pipeline design process investigates the impact on the RUL estimate when the best 10 models are selected from a random search of 10 (90 unique models based on 10 random hyperparameter initialisations for each of the original authors 9 loss functions) and a random search of 100 (900 unique models),

4.3.6. Model Choice

The original author utilises NNs in their study which are black box and computationally expensive. This can undermine the operators trust in the chosen analytic as outputs can not be explained by the model, increasing the risk associated with incorporating model suggestions into decision making processes. Linear Regression (LR) models reside at the other end of the model spectrum as they are cheap to train and simple to understand. However, NNs are able to tackle complex data problems with complicated underlying relationships

Pipeline Stage	Parameter Settings
Dataset	IMS or FEMTO dataset
Sensor channel	Horizontal or Vertical aligned
Subsampling	Lose 1/8, 1/4, 1/2 or no data
Spectrogram Bins	10, 20 or 40 bins
Hyperparam. Opt.	Random search of 10, or 100
Model Choice	NNs or Linear Regression (LR)

Table 3. Summary of pipeline stages and parameters

which cannot be captured by the LR model. In this stage of the pipeline design process, the chosen models are NNs and LR models to compare the RUL prediction between computationally expensive, sophisticated models and interpretable, low computation models.

4.3.7. Evaluating Uncertainty

The original data for each dataset was processed to remove every 8th, 4th or 2nd data point for every datafile in the dataset and resaved; and this process was repeated for each sensor channel. This ensured all combinations of dataset, data sampling and sensor channel were available to train the models. Each model type was trained on all combinations of dataset, data sampling and sensor channel, with the data preprocessed for each selected bin count. For each of these combinations, the NN model hyperparameters were chosen with a random search of 10 or 100, with the model and metrics saved for later processing. The metrics chosen to validate the models were R^2 , mean squared error (MSE), RMSE, mean squared log error (MSLE), root mean squared log error (RMSLE), in line with those chosen by (Hahn & Mechefske, 2022). The conditions for successful models to be progressed to the testing stage were a training (and for NN models, validation) performance of $R^2 > 0.2$ and $RMSE < 0.35$, which was applied again after the testing stage to shortlist the top models. To obtain the quantiles, the testing data was run through each of the top models to obtain their RUL predictions, where the 5 %, 25 %, mean, 75 % and 95 % percentiles were calculated for each timestep. The choice of testing data was Run 1, Bearing 4 for IMS and Bearing 1_3 for FEMTO, as the original authors method performed well on these and was decided to be a good point of comparison. This process generated results for all combinations of the 2 datasets, 2 sensor channels, 4 data sampling regimes, 3 spectrogram bin counts, 2 hyperparameter optimisation searches and 2 model choices, resulting in 192 distinct pipeline designs. For each pipeline, the maximum number of models to analyse is the top 10 NNs and a LR model, however not all combinations produced this amount of models that successfully passed the metric bounding criteria.

5. RESULTS

As mentioned in section 4.3.7, the case for comparison between (Hahn & Mechefske, 2022) and this work was Run 1, Bearing 4 testing data from IMS and Bearing 1_3 testing data

for FEMTO.

5.1. IMS Results

The RUL prediction shown in figure 2 shows (Hahn & Mechefske, 2022) results for their best performing model on the IMS dataset. This NN model has a Weibull-RMSE Combined loss function, 4 layers with 32 units per layer, 0 % dropout probability, lambda of 0.53, Weibull shape parameter (β) of 2 and characteristic lifetime (η) of 63.9 days. In figure 2, the bearing lifetime extends from 0 % to 100 %, where the jumps are due to the gaps in data collection from the original IMS experiment. The NN predictions are smoothed using a 2 hour rolling average to more clearly demonstrate the trends in the prediction. As shown, the model fits this data well, with a low RMSE score of 0.146, and a high R^2 score of 0.735.

Figure 3 shows the quantiles and mean RUL estimate from the top NN models across all IMS pipelines which met the training and validation metric bounding criteria. The quantiles are calculated on the models performance on Run 1 Bearing 4 testing data from the IMS dataset and the mean of these predictions result in a R^2 of 0.355 and RMSE of 0.228. From approximately 50 % bearing lifetime the quantiles bound the actual lifetime percentage until failure, with the mean fitting the true lifetime percentage well from 60 % lifetime onwards. As shown, the models do not predict early-mid life with any success, which may mislead an operator incorporating the model into a maintenance decision as the model cannot distinguish between any states $< 50\%$ lifetime. Some of this deviation may be explained by the large jumps in lifetime % within the first 10 days of the experiment, compared to the much smoother data collection from day 15 to failure, regardless, this still undermines confidence in the predictions.

The results for the IMS LR models are shown in figure 4. While the quantiles bound the true lifetime from experiment start to end, the lack of incorporated knowledge allows the models to expand to many multiples of bearing lifetime and into negative values. This results in a mean R^2 score of -0.223 and RMSE of 0.314, despite all of the models successfully meeting the R^2 and RMSE bounds in the training stage. Additionally, the RMSE for the testing results is still within (Hahn & Mechefske, 2022) 0.35 boundary while producing unreliable predictions, suggesting other forms of validation are required in tandem to discount unsuitable models. This demonstrates that applying regression models that minimise computational cost or maximise interpretability cannot always perform the required task, and further demonstrates the need for hybrid modelling approaches to incorporate known behaviour.

The pipeline design parameter summary is shown in table 4 which shows the breakdown of pipeline stage parameter counts in the final model selection. The maximum number of models is the top 10 NNs from the 46 IMS NN pipelines,

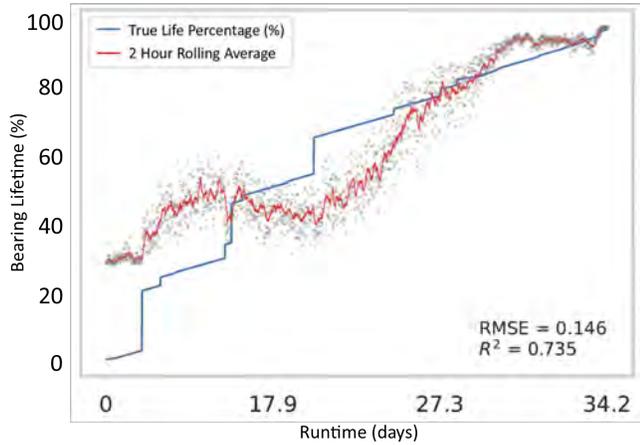


Figure 2. IMS Run 1, Bearing 4 Test Results ((Hahn & Mechefske, 2022)). $R^2 = 0.735$, $RMSE = 0.146$

and single LR model for each of the 46 IMS LR pipelines if all of these models trained successfully. This results in an acceptance rate of 79.8 % for the NNs (367 out of potential 460 models were successful) and only 39.1 % for the LR (18 out of potential 46 models were successful), demonstrating that the NN is more likely to be successful at this prognostic task. For the 18 successful LR models, the sensor alignment choices are split evenly, implying the sensor orientation neither hindered nor helped the models performance, while the NNs tended to favour the horizontal channel as chosen by (Hahn & Mechefske, 2022). Interestingly, for the sampling regime the LR models favoured learning from the least data and fared equally amongst the other options. The NNs were also fairly evenly spread amongst the sampling options, favouring the maximum amount of data. The LR models selected the most condense spectrogram the least, implying the higher dimensional representations provided more useful degrees of freedom to the model. Conversely, the NNs were more evenly spread across the bin options, suggesting all options provided the NNs with enough information. To summarise, it appears that on the IMS dataset, the most influential design parameter was the dimensionality of the input data for the LR models as shown by the aversion to the 10 bin spectrogram, and the time available to optimise the hyperparameters for the NN as this displayed the largest diversion by model contribution in favour of larger number of searches.

5.2. FEMTO Results

The RUL prediction shown in figure 5 shows (Hahn & Mechefske, 2022) results for their best performing model on the FEMTO dataset, with a Weibull only RMSLE loss function, 2 layers with 32 units per layer, 0.25 % dropout probability, lambda of 2.28, Weibull shape parameter (β) of 2 and characteristic lifetime (η) of 4.8 hours. The trend of the predictions is shown by a 2-minute rolling average with straight line from 0

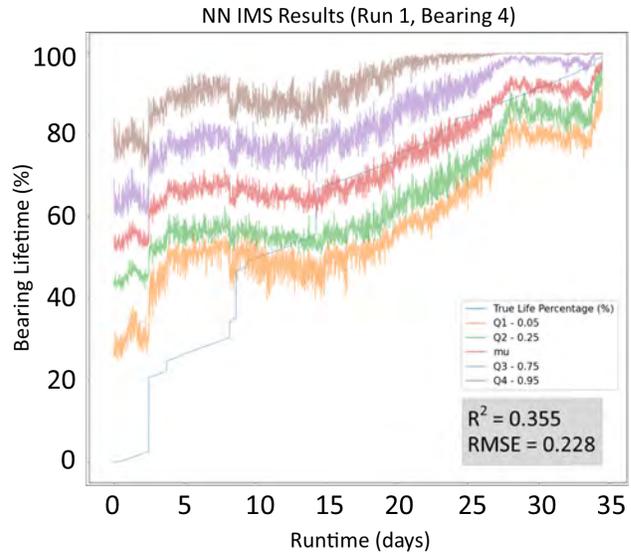


Figure 3. IMS Run 1, Bearing 4 Test Result Uncertainty (NN Model). $R^2 = 0.355$, $RMSE = 0.228$

Pipeline Stage	Value	NN	LR
Max Models	NN - 460	367	-
	LR - 46	-	18
Sensor	Horizontal	214	9
	Vertical	153	9
Sampling	Normal	101	4
	- 1/8	95	4
	- 1/4	86	4
	- 1/2	85	6
Spec.Bins	10	126	2
	20	127	8
	40	114	8
HyperParam Search	10	137	-
	100	230	-

Table 4. Summary of IMS pipeline settings for LR and NN models (by successful model counts)

- 100 % demonstrating the bearing lifetime. This NN fits the data well as shown by the low RMSE of 0.133 and high R^2 of 0.788.

The NN FEMTO uncertainty plot is shown in figure 6, which shows the quantiles bounding the whole bearing lifetime, but does not narrow as much as the IMS results at end of life. This larger spread in predictions demonstrates the volatility of the NN predictions on this dataset, as depending on the model, the prediction could be anywhere between 0 and 60 % at start of life and 50-100 % at end of life. The mean prediction has R^2 of 0.729 and RMSE of 0.15 which suggest the mean has a decent fit, however, it can be seen that the models tend to overestimate degradation early-mid life and underestimates mid-end life. If used to inform maintenance schedules, the start of life predictions could result in actions being taken too early where still usable components are prematurely replaced.

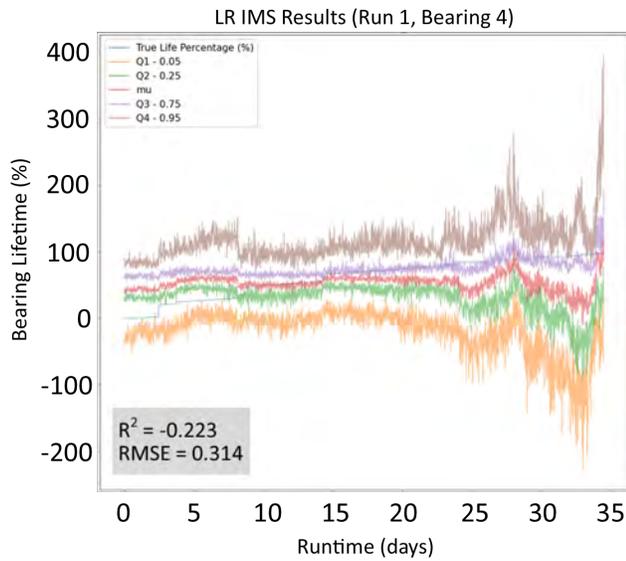


Figure 4. IMS Run 1, Bearing 4 Test Result Uncertainty (LR Model). $R^2 = -0.223$, $RMSE = 0.314$

Actions taken based upon the end of life predictions could be left too late, putting operators at risk of unplanned outages.

The fit of the LR models in figure 7 shows consistent estimations early-mid life, then a huge divergence of multiple lifetimes in positive and negative direction is observed in the final stages of the bearing life. This may be due to the rapid decay of the bearings, as the spectrograms show a rapid increase in vibration for some of the training data in the later stages of the experiment. As the end of life prediction is arguably the most crucial aspect of prognostics, these LR models could be considered a risk for any operator to employ in maintenance activities.

In the pipeline design summary in table 5, both the NN and LR models have relatively even contributions to the 198 successful NN models and 24 LR models from all settings for the sampling and spectrogram bin options, suggesting these do not have a great influence on the model performance. This is also true for the sensor alignment for the LR models, while for the NN models there is almost entirely self selected horizontal channel, as in (Hahn & Mechefske, 2022). This suggests that the horizontal sensor provides the most useful information for the NN model. Additionally, the NN has strong contribution from the larger hyperparameter search with a majority of models being chosen by the random search of 100. Finally the NN models have an acceptance rate of 43.0 % while the LR models have an acceptance rate of 52.2 %. Interestingly, while the mean NN performance produces better results for R^2 and RMSE, the LR models are more consistently performing above the set metric boundaries and being accepted into the testing stage. Despite their unsuitable design, the choice of metrics and bounds used to assess these

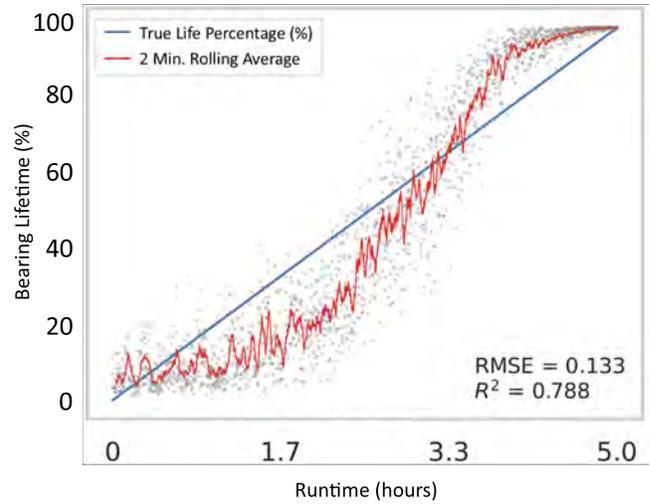


Figure 5. FEMTO Bearing 1_3 Test Results ((Hahn & Mechefske, 2022)). $R^2 = 0.788$, $RMSE = 0.133$

Pipeline Stage	Value	NN	LR
Max Models	NN - 460	198	-
	LR - 46	-	24
Sensor	Horizontal	197	12
	Vertical	1	12
Sampling	Normal	45	6
	- 1/8	43	6
	- 1/4	55	6
	- 1/2	55	6
Spec.Bins	10	69	8
	20	63	8
	40	66	8
HyperParam Search	10	77	-
	100	121	-

Table 5. Summary of FEMTO pipeline settings for LR and NN models (by successful model counts)

models suggest they should be accepted, again suggesting that models require more diverse validation to determine their general suitability, or what situations they may be best suited for. This may also require an appreciation of the similarity of the training and testing data, as models that succeed at the training stage should be trusted to succeed in the testing or online monitoring stage.

This sensitivity analysis has demonstrated that the approach taken to data pipeline definition can have a significant impact on the accuracy of prognostic algorithms, with evidence for a specific bearing vibration case-study provided. This case-study suggests that when developing a data pipeline for this purpose valid models can be selected from a variety of plausible data pipeline configurations while resulting in a diverse range of learned RUL trajectories.

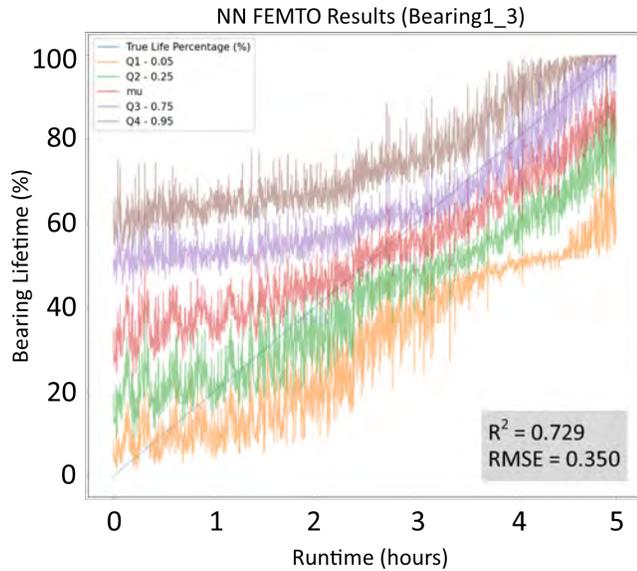


Figure 6. FEMTO Bearing 1_3 Test Result Uncertainty (NN Models). $R^2 = 0.729$, $RMSE = 0.150$

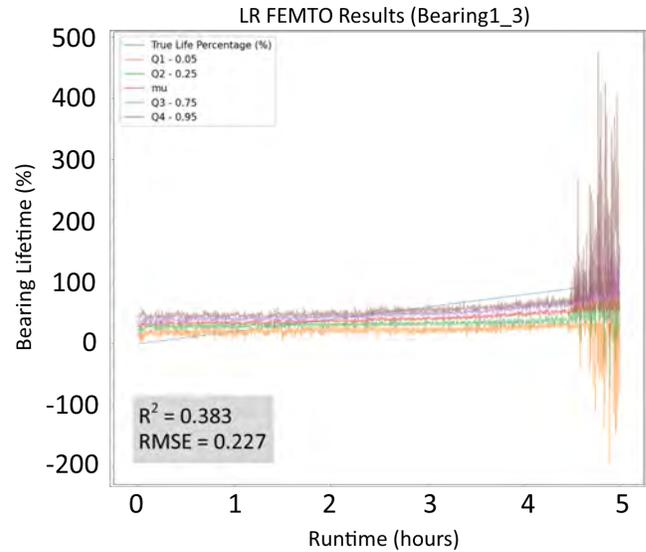


Figure 7. FEMTO Bearing 1_3 Test Result Uncertainty (LR Models). $R^2 = 0.383$, $RMSE = 0.227$

6. CONCLUSION AND FUTURE WORK

Civil nuclear is a safety critical industry which cannot readily deploy data-driven analytics in decision-making processes without quantification of the uncertainties involved. Consequently, in this work an analysis of the impact of data acquisition pipeline design decisions on the performance of an existing hybrid RUL model for bearing prognostics was conducted. It was shown that the design decisions made at key stages of the data acquisition pipeline can create a large variance of potential RUL trajectories for both NN and LR models on both of the bearing run-to-failure datasets utilised in the study. The models were more sensitive to some design decisions than others, such as the available number of hyperparameter optimisation searches for the NN or the dimensionality of the input features for the LR model (on the IMS dataset). The presence of incompatible design decisions was not suggested by the results as many stages produced an equal number of successful models across the different design options. This suggests that valid models could be generated from completely different pipeline designs, which result in an entirely different learned RUL trajectory. Understanding how the data acquisition pipeline can impact on hybrid prognostic tools can allow nuclear plant operators to justify utilising resources towards reducing high uncertainty areas in the pipeline design to provide more confidence in applying these tools to support maintenance processes. This is of particular concern in the nuclear industry as ML algorithms applied to rotating plant deployed in nuclear engineering environments experience unique operating conditions, such as legacy data acquisition systems that have been upgraded over time without emphasis on the data that will be used for ML purposes.

The models were filtered by a requirement of $R^2 > 0.2$ and $RMSE < 0.35$ to remove unsuitable models before progressing to the testing stage, as in (Hahn & Mechefske, 2022). The results showed that the chosen metrics are not sufficient to definitively identify unsuitable models and are not descriptive enough to show the operator where model application should and should not be trusted. Additionally, the chosen training and testing data may not have been sufficiently comparable for LR type models, as shown by models that had been deemed acceptable in the training stage performing poorly on IMS testing data in figure 4.

To further develop this work, more analysis would be conducted on the impact of metric bias in the model selection process. Models were selected and ranked based on their R^2 and RMSE scores, but a different selection of shortlisted models may have been generated if different metrics had been used or prioritised. Additionally, if it was discovered that some models were more accurate for end of life predictions while other models are more suited for early-mid life, this may not be captured by summary statistics used to qualify the overall model usefulness. Additional methods to describe where the model is successful is needed to further justify the models use for specific prognostic stages, which could be aided by the application of explainability tools. Finally, for a more robust comparison, knowledge would be incorporated into different model types. This would provide more hybrid combinations to compare against, while investigating how model bias impacts the RUL prediction.

REFERENCES

- Abernethy, R. B. (2004). The new weibull handbook : reliability and statistical analysis for predicting life, safety, supportability, risk, cost and warranty claims..
- Baur, M., Albertelli, P., & Monno, M. (2020, 03). A review of prognostics and health management of machine tools.. doi: 10.1007/s00170-020-05202-3
- Canada Nuclear Safety Commission. (2012, November). *Maintenance programs for nuclear power plants*. Regulatory Document, RD/GD-210. (Online: nuclearsafety.gc.ca)
- Coble, J., Ramuhalli, P., Bond, L., Hines, J., & Upadhyaya, B. (2015, 07). A review of prognostics and health management applications in nuclear power plants. *International Journal of Prognostics and Health Management*, 6, 1-22. doi: 10.36001/ijphm.2015.v6i3.2271
- Goebel, K., Eklund, N., & Bonanni, P. (2006, 01). Fusing competing prediction algorithms for prognostics. In (Vol. 2006, p. 10 pp.). doi: 10.1109/AERO.2006.1656116
- Hahn, T. V., & Mechefske, C. K. (2022). *Knowledge informed machine learning using a weibull-based loss function*. Journal of Prognostics and Health Management. (Preprint available: <https://doi.org/10.48550/arXiv.2201.01769>, code available: <https://github.com/tvhahn/weibull-knowledge-informed-ml>)
- Hart, E. (2018). *Wind turbine dynamics identification using gaussian process machine learning* (Unpublished doctoral dissertation). University of Strathclyde.
- Jammu, N., & Kankar, P. (2011, 10). A review on prognosis of rolling element bearings. *International Journal of Engineering Science and Technology*, 3.
- JCGM Working Group 1. (2008, 09). *Evaluation of measurement data — Guide to the expression of uncertainty in measurement* (Tech. Rep.). JCGM.
- Kennedy, M. C., & O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society Series B*, 63(3), 425–464.
- Kumar, S., Mukherjee, D., Guchhait, P., Banerjee, M. R., Srivastava, A. K., Vishwakarma, D., & Saket, R. (2019, 07). A comprehensive review of condition based prognostic maintenance (cbpm) for induction motor. *IEEE Access*, 7, 90690-90704.
- Lee, J. , Qiu, H. , Lin, J. and Rexnord Technical Services. (2007). IMS, University of Cincinnati. "Bearing Data Set", NASA Ames Prognostics Data Repository. (<http://ti.arc.nasa.gov/project/prognostic-data-repository>, NASA Ames Research Center, Moffett Field, CA)
- Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to rul prediction. *Mechanical Systems and Signal Processing*, 104, 799-834. doi: <https://doi.org/10.1016/j.ymsp.2017.11.016>
- Liao, L., & Köttig, F. (2014). Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. *IEEE Transactions on Reliability*, 63(1), 191-207. doi: 10.1109/TR.2014.2299152
- NASA Ames Prognostics Data Repository. (2012). *Femto bearing data set*. NASA Ames Research Center, Moffett Field, CA. (<http://ti.arc.nasa.gov/project/prognostic-data-repository>)
- Nisbet, R., Elder, J., & Miner, G. (2009). Chapter 8 - advanced algorithms for data mining. In R. Nisbet, J. Elder, & G. Miner (Eds.), *Handbook of statistical analysis and data mining applications* (p. 151-172). Boston: Academic Press. doi: <https://doi.org/10.1016/B978-0-12-374765-5.00008-5>
- Ramuhalli, P., Walker, C., Agarwal, V., & Lybeck, N. J. (2020). *Development of prognostic models using plant asset data* (Tech. Rep.). Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States); Idaho National . . .
- Rexnord Industries, LLC, Gear Group. (n.d.). Failure analysis gears-shafts-bearings-seals [Computer software manual].
- Yung, C., & Bonnett, A. (2004). Repair or replace? *IEEE Industry Applications Magazine*, 10(5), 48-58. doi: 10.1109/MIA.2004.1330770
- Zhao, X., Kim, J., Warns, K., Wang, X., Ramuhalli, P., Cetiner, S., . . . Golay, M. (2021). Prognostics and health management in nuclear power plants: An updated method-centric review with special focus on data-driven methods. *Frontiers in Energy Research*, 9. doi: 10.3389/fenrg.2021.696785

Data-driven Prognostics based on Evolving Fuzzy Degradation Models for Power Semiconductor Devices

Khoury Boutrous¹, Iury Bessa², Vicenç Puig¹, Fatiha Nejjari¹, Reinaldo M. Palhares³

¹ *Advanced Control Systems, Technical University of Catalonia (UPC), Rambla Sant Nebridi 22, 08222 Terrassa, Spain.
boutrous.khoury, vicenc.puig, fatiha.nejjari @upc.edu*

² *Federal University of Amazonas, Department of Electricity, Manaus, Brazil
iurybessa@ufam.edu.br*

³ *Federal University of Minas Gerais, Department of Electronics Engineering, Belo Horizonte, Brazil
rpalhares@ufmg.br*

ABSTRACT

The increasing application of power converter systems based on semiconductor devices such as Insulated-Gate Bipolar Transistors (IGBTs) has motivated the investigation of strategies for their prognostics and health management. However, physics-based degradation modelling for semiconductors is usually complex and depends on uncertain parameters, which motivates the use of data-driven approaches. This paper addresses the problem of data-driven prognostics of IGBTs based on evolving fuzzy models learned from degradation data streams. The model depends on two classes of degradation features: one group of features that are very sensitive to the degradation stages is used as a premise variable of the fuzzy model, and another group that provides good trendability and monotonicity is used for the auto-regressive consequent of the fuzzy model for degradation prediction. This strategy allows obtaining interpretable degradation models, which are improved when more degradation data is obtained from the Unit Under Test (UUT) in real time. Furthermore, the fuzzy-based Remaining Useful Life (RUL) prediction is equipped with an uncertainty quantification mechanism to better aid decision-makers. The proposed approach is then used for the RUL prediction considering an accelerated aging IGBT dataset from the NASA Ames Research Center.

ACRONYMS

RUL	Remaining Useful Life
EOL	End of Life
PHM	Prognostics and Health Management
CM	Condition-based Monitoring

Boutrous Khoury et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

TS	Takagi-Sugeno
UUT	Unit Under Test
RLS	Recursive Least Squares
SFWRLS	Sliding-windowed Fuzzily Weighted Recursive Least Squares
MAPE	Mean Absolute Percentage Error
RA	Relative Accuracy
IGBT	Insulated-Gate Bipolar Transistor
C-trig	Cummulative trigonometric function
$V_{CE_{on}}$	On-state Collector-Emitter Voltage

1. INTRODUCTION

The IGBT has long established itself as a competent successor to prior power semiconductors such as the power bipolar junction transistor (BJT), Darlington transistor, and metal oxide semiconductor field-effect transistor (MOSFET). It functions by combining the desirable properties of a high input impedance and high switching speeds of the MOSFET with the low saturation voltage of the BJT, enabling a voltage-controlled transistor that is capable of containing large collector-emitter currents with a virtually zero-gate current drive. The product is a transistor variant that offers medium to high power application abilities, low ON-resistance, and fast switching compared to its predecessors.

As any component in a system, IGBTs are prone to failure under certain operating conditions, primarily from electrical and thermal stress caused by conditions such as high temperature and cycling effects (Lu & Sharma, 2009). However, the critical nature of power semiconductors in the chain of operation of most systems may cause a total shutdown emanating from an otherwise inexpensive source. From an industrial survey by (Yang et al., 2011), the majority of respondents indeed assert that power electronic devices are one of

the most fragile components in most industries and the need for increased research interest in reliability monitoring and improvement. This monitoring is essential, especially in critical systems such as aviation, where the neglecting cost may be more than just monetary. In lieu of this, there is a need for reliable and robust diagnostic and prognostics techniques that seek to avert to a low degree any spontaneous call for maintenance that introduces unplanned expenditures and manages faults or deterioration during inception before they escalate to disruptive levels. Using a model-based method, a failure precursor's features or both, a fault source can be detected, isolated, and the RUL of an IGBT predicted for maintenance responses such as planned replacements undertaken at an optimal time before its End of Life (EOL).

Some common failure modes in IGBTs are the gate diode degradation, body diode degradation, the bond wire and solder layer fatigues (Nguyen & Kwak, 2020). For appropriate health management algorithms, it is desirable to study a component's observable parameters that conspicuously show deviation from their normal behaviour reflecting an associated anomaly, i.e., a failure mode when in operation. For instance, during failure modes such as the bond wire and solder layer fatigues, there is an associated increase in measured On-state Collector-Emitter Voltage ($V_{CE_{on}}$), which results from an increased bond wire resistance for bond wire fatigues and thermal resistance associated with the latter due to the lack of effective heat dissipation between adjacent layers. The transistor turn off time have also been identified as a parameter of interest for latch-up faults in (Brown et al., 2010). These parameter-failure mode pairing are acquired through a procedure termed FMMEA (Failure Modes, Mechanisms and Effects Analysis) under accelerated aging procedures. The criteria of choosing a specific prognostics parameter depends on its sensitivity to the failure mode and also the ease of attaining accurate measurements from sensors. For example, the junction temperature as a precursor is indicative of most thermal failure modes, but the difficulty in sensor integration during pre-designs as well as inaccurate measurements limits its applicability. Therefore, works such as (Eleffendi & Johnson, 2016) considers the junction temperature as a failure precursor, but obtained through a lookup table considering measured $V_{CE_{on}}$. With appropriately measured precursors from IGBTs, different prognostics procedures have been studied in literature.

In (Saha, Celaya, Wysocki, & Goebel, 2009), a model-based prognostics procedure using a particle filter is used based on a fitted model on the collector-emitter leakage current obtained from an accelerated aging procedure. However, in (Haque, Choi, & Baek, 2018) an auxiliary particle filter proved to have a better variance and robustness of RUL predictions compared to particle filters using the $V_{CE_{on}}$ as a failure precursor, a predominant choice in most papers. Data-based algorithms have also been extensively studied, both statistically

and in the area of artificial intelligence. Statistically, for a data-based prognostics of IGBTs, (Ismail, Saidi, Sayadi, & Benbouzid, 2019) employed the Gaussian process regression, whilst later in (Ismail, Saidi, Sayadi, & Benbouzid, 2020) the authors used a modified maximum likelihood method to predict the RUL. The results show that the Gaussian process regression has better prognostics metrics than the modified maximum likelihood method. With the $V_{CE_{on}}$ as a chosen precursor in (Ahsan, Stoyanov, & Bailey, 2016), Neural Network (NN) and Adaptive Neuro Fuzzy Inference System (ANFIS) models are used to predict the RUL, the NN showed better performance compared to the ANFIS. In (Alghassi, Perinpanayagam, & Samie, 2016), the authors proposed a time delay neural network algorithm in tandem with a probabilistic function with $V_{CE_{on}}$ as the precursor parameter, which proved to be more efficient than a stand alone NN model. Comprehensive reviews exist in literature on the broad subject, encompassing the type of failures (Nguyen & Kwak, 2020; Hanif, Yu, DeVoto, & Khan, 2019), precursor parameter attainment (Zhang, Liu, Li, & Li, 2020) and prognostics methods (Degrenne, Kawahara, & Mollov, 2019; Kabir, Bailey, Lu, & Stoyanov, 2012) employed on power semiconductors in general.

Although adaptive prognostics methods are able to modify their parameters according to the data stream behavior to reduce the modeling error, their structure are fixed and there is no clear relationship between their evolving degradation stage and their parameters (Angelov, 2012). Otherwise, evolving systems are known by their ability of modifying both parameters and structure to provide explainable representations for data-streams. While their parameters are adapted to minimize the modeling error, the structure becomes more complex to represent novel dynamics which can be related to the achievement of novel degradation stages in prognostics problems. Recently, evolving fuzzy degradation models are proposed for aiding data-stream-driven Prognostics and Health Management (PHM) systems (Camargos, Bessa, D'Angelo, Cosme, & Palhares, 2020; Camargos et al., 2021; Ahwiadi & Wang, 2022). In particular, those models are used to capture the degradation dynamics and predict the equipment RUL. In this regard, evolving prognostic approaches have been successfully applied to ball bearings (Camargos et al., 2020) and lithium-ion batteries providing (Camargos et al., 2021; Ahwiadi & Wang, 2022) competitive results with some interpretability features. This motivates the application of those methods for the challenging IGBT prognostic problem.

Properties such as monotonicity and trendability of a chosen extracted feature or dimensionally reduced subspace of selected features is a strong prerequisite for attaining a good RUL prediction, employed in prognostics algorithms. However, the downside of this procedure is that the granularity of the degradation data is diminished or lost when smoothing tools are applied to attenuate these properties. This result

in algorithms that sacrifice interpretability for improved RUL predictions. Even though it is agreed that the primary end goal of prognostics algorithms is to improve the RUL prediction, there must be a motivation to consider characteristics of the degradation trend, which may be used for secondary purposes or aid in a better RUL prediction. This especially comes in handy when considering degradation data as used in this paper, a stage-based degradation process, where classification of the stages may prove to be important for better estimating the RUL.

The approach in this paper considers a data-based evolving fuzzy model that uses two classes of input features: an interpretable feature as a premise variable and a RUL prediction-friendly counterpart in the autoregressive consequent of the fuzzy model. In particular, the degradation representation is based on the Evolving Ellipsoidal Fuzzy Information Granules (EEFIG), whose has already been applied for clustering (Cordovil, Coutinho, Bessa, D’Angelo, & Palhares, 2020), fault diagnosis (Cordovil et al., 2020), and leaning-based control (Cordovil, Coutinho, Bessa, Peixoto, & Palhares, 2022) approaches. The merit of this methodology over others is that it provides a platform with a dual function mode: (1) Providing a good RUL prediction in conjunction with (2) classifying the different stages of degradation, as shown in Figure 1, enabling interpretability.

2. IGBT AGING AND DEGRADATION

As stated in the prequel, for prognostics, it is imperative to acquire parameters that explicitly represent the failures, such that a study of their behaviour can inherently constitute a knowledge of the failure mechanism. To measure all useful related precursor parameters, the IGBTs are subject to aggressive thermal or electrical cycles of stress in an experimental environment until a failure happens. In this work, a run-to-failure experiment on 4 IGBTs undertaken by (Sonnenfeld, Goebel, & Celaya, 2008) is considered. The experiment involves subjecting power transistor devices to DC square wave signals at the gate, placing the devices under thermal stress. This aging process is undertaken until a latch-up or thermal runaway (EOL) when signals are switched steadily between 0V and 4V, with temperature controlled between 329°C and 330°C outside the rated temperature of the test transistors. The transient data collected when the devices switch are the (i) Collector-emitter turn on Voltage; (ii) Gate Voltage; and (iii) Collector current.

3. DATA-DRIVEN PROGNOSTICS BASED ON EVOLVING FUZZY DEGRADATION MODEL

3.1. Degradation features extraction and selection

From the literature, the $V_{CE_{on}}$ is the predominantly chosen precursor, which has proven its efficacy as well as practicability compared to other parameters, with various pre-evaluated

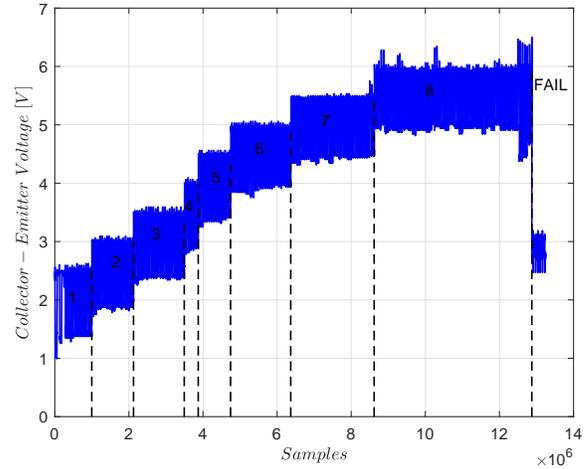


Figure 1. Measured collector-emitter voltage from aging test of IGBT1 showing stages of degradation.

metrics supporting its selection. Therefore, the $V_{CE_{on}}$ is selected as the parameter of interest in this paper. With $V_{CE_{on}}$ as the selected Condition-based Monitoring (CM) data, features are extracted serving as a pseudo-representation of the degradation behaviour. The raw data, shown in Figure 2, are almost always noisy, an undesirable characteristic for a RUL prediction. These features, either frequency or temporal based, must exhibit desirable characteristics that ensures accurate RUL extrapolations with less uncertainty (Gouriveau, Medjaher, & Zerhouni, 2016). Two types of characteristics of input data into the proposed algorithm are considered. First, a feature that satisfies the traditional desirable properties of monotonicity, trendability and prognosticability for an accurate and less uncertain RUL prediction is considered and a feature that represents the shape of degradation showing the different stages. Unlike the first case, the accuracy of the RUL is not deemed a factor. Thus, for the autoregressive consequent feature, a feature construction from (Javed, Gouriveau, Zerhouni, & Nectoux, 2015) is considered. The authors employ the standard deviation (SD) of Cummulative trigonometric function (C-trig) on the data set. This was proven to possess overall better prognostics characteristics backed with more accurate RUL compared to generic features when tested on a case study. Two C-trigs, as proposed in (Javed et al., 2015), are considered and the best selected based on the suitability metric (1), as proposed in (Celaya, Saxena, Saha, & Goebel, 2011).

$$\text{Suitability} = \begin{bmatrix} \text{Monotonicity} \\ \text{Trandability} \\ \text{Prognosticability} \end{bmatrix}^T \begin{bmatrix} 1 \\ 0.976 \\ 1 \end{bmatrix} \quad (1)$$

For the premise variable of the fuzzy model, features of the mean and the root mean square is considered for selection presented in Table 2. Smoothing is done with the moving

average and the window length selected equal to the number of samples in each testsets performed on individual IGBTs. Considering a data-stream $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^n$ the selected features are presented in Table 1.

Table 1. Trigonometric features for the premise variable.

Feature	Formula
SD of asinh(X)	$\sigma \left(\log \left[x_i + (x_i^2 + 1)^{\frac{1}{2}} \right] \right)$
SD of atan(X)	$\sigma \left(\frac{i}{2} \log \left(\frac{i+x_i}{i-x_i} \right) \right)$

Table 2. Features for the autoregressive consequent variable.

Feature	Formula
Energy	$\sum_{i=1}^n E(x_i)$
Root Mean Square (RMS)	$\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}$

The cumulative function as from (Javed et al., 2015), is done by considering a simultaneous point-wise running total and scaling of a time series:

$$CF_i(X) = \frac{\sum_{i=1}^n X(i)}{|\sum_{i=1}^n X(i)|^{\frac{1}{2}}} \quad (2)$$

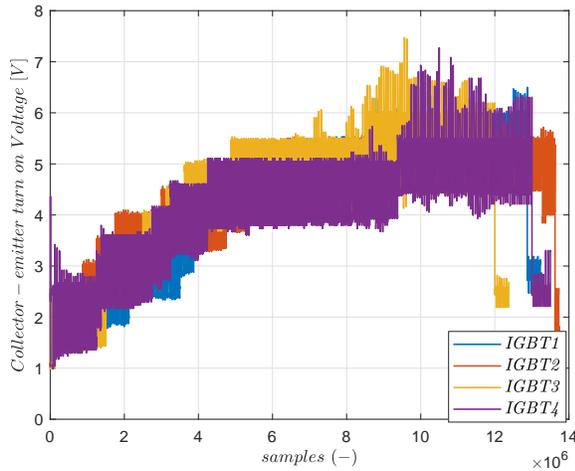


Figure 2. Measured collector-emitter voltage from aging test of 4 IGBTs.

Feature selection. For the auto-regressive consequent feature of the fuzzy model, the *cummulative SD of (atan)* is selected, with a suitability score of 2.972 compared to 2.957 of (*asinh*). For the premise variable, the RMS feature was

selected based on the best results from the two features (i.e Table 2) as inputs.

3.2. Evolving Ellipsoidal Fuzzy Information Granules

In (Cordovil et al., 2020), the Evolving Ellipsoidal Fuzzy Information Granules (EEFIG) model and its evolving granular learning algorithm are introduced. The learning algorithm is an online data processing that employs evolving fuzzy information granules based on the parametric principle of justifiable granularity (Pedrycz & Wang, 2016). In this paper, we propose employing the EEFIG algorithm to model the degradation of the IGBTs.

An EEFIG is a collection of N granules $\mathbb{G}_k = \{\mathcal{G}_k^1, \dots, \mathcal{G}_k^N\}$, where each granule is a fuzzy set $\mathcal{G}_k^i = (\mathbb{R}^{n_z}, g_k^i)$, where $g_k^i : \mathbb{R}^{n_z} \rightarrow [0, 1]$ is the membership function of the EEFIG \mathcal{G}_k^i . The membership function ω_k^i is parameterized by the granular prototype \mathcal{P}_k^i of the i -th granule at the time instant k , which is also a numerical evidence basis for the granulation process. The granule prototype is defined as follows:

$$\mathcal{P}_k^i = \left(\underline{\mu}_k^i, \mu_k^i, \bar{\mu}_k^i, \Sigma_k^i \right), \quad (3)$$

where $\underline{\mu}_k^i$, μ_k^i and $\bar{\mu}_k^i$ are the lower, mean and upper bound vectors of the i -th EEFIG at time k and Σ_k^i is the inverse of its covariance matrix. Given the granule prototype \mathcal{P}_k^i , the membership function of an EEFIG is parameterized as

$$\omega_k^i(z_k) = \exp \left\{ - \left[(z_k - \mu_k^i)^\top (\Delta_k^i)^{-1} (z_k - \mu_k^i) \right]^{1/2} \right\}, \quad (4)$$

where, for $p \in \mathbb{N}_{\leq n_z}$,

$$\Delta_k^i = \text{diag} \left\{ \left(\frac{\bar{\mu}_{k,1}^i - \underline{\mu}_{k,1}^i}{2} \right)^2, \dots, \left(\frac{\bar{\mu}_{k,p}^i - \underline{\mu}_{k,p}^i}{2} \right)^2 \right\},$$

being $\bar{\mu}_k^i$, and $\underline{\mu}_k^i$ the semi-axes of the i -th EEFIG prototype such that $\underline{\mu}_k^i < \mu_k^i < \bar{\mu}_k^i$ (Wang, Shi, Wang, & Zhang, 2014). The normalized membership functions g_k^i at the k -th time instant for i -th granule is

$$g_k^i(z_k) = \frac{\omega_k^i(z_k)}{\sum_{i=1}^N \omega_k^i(z_k)}. \quad (5)$$

Moreover, the distance of a given data sample $z_k \in \mathbb{R}^{n_z}$ to the i -th EEFIG is given by the square of Mahalanobis distance:

$$d(z_k, \mu_k^i) = (z_k - \mu_k^i)^\top \Sigma_k^i (z_k - \mu_k^i). \quad (6)$$

The granulation process is the updating of the EEFIG model-based on the data stream. The updates are performed aiming at improve the so-called granular performance index with re-

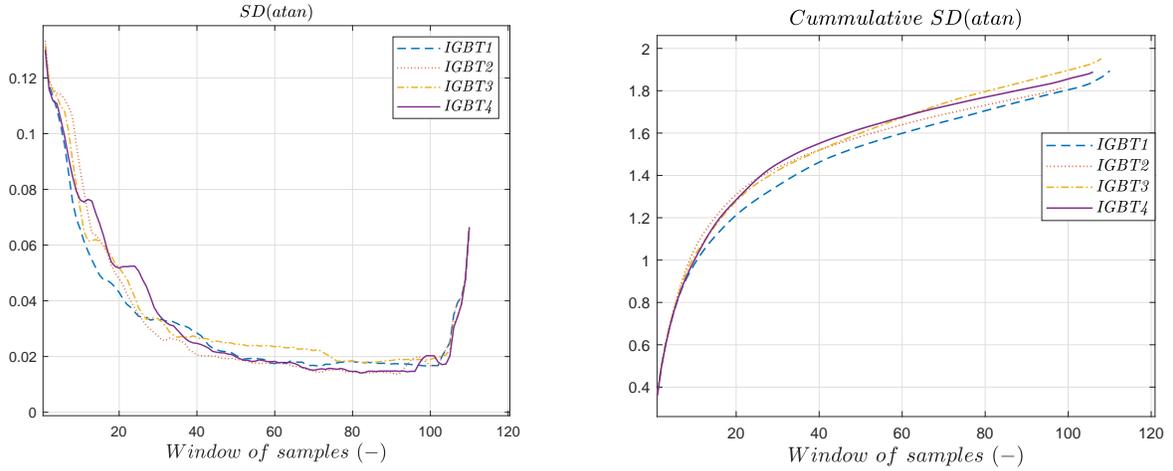


Figure 3. Selected auto-regressive consequent feature of the 4 IGBTs. (Left.) SD(atan) (Right.) C-SD(atan).

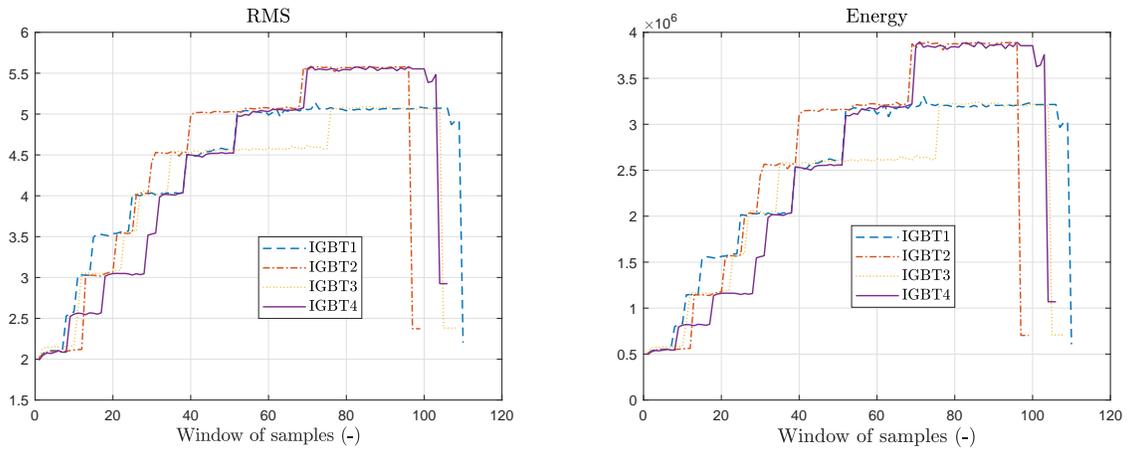


Figure 4. Considered premise features of the 4 IGBTs.

spect to a data sample. The performance index of the i -th granule with respect to the sample z_k , denoted \bar{Q}_k^i , is defined as

$$\bar{Q}_k^i(z_k) = d(z_k, \mu_k^i) |G_k^i| \quad (7)$$

where $|\cdot|$ is the fuzzy cardinality operator of the i -th EEFIG, whose update is performed as follows

$$|G_k^i| = |G_{k-1}^i| + g_k^i(z_k) - \frac{\partial g_k^i(z_k)}{\partial \mathcal{P}_k^i}, \quad (8)$$

where the term $\frac{\partial g_k^i(z_k)}{\partial \mathcal{P}_k^i}$ is computed as described in (Cordovil et al., 2022). The total EEFIG performance index is the sum of the data sample contribution index of each granule:

$$Q_k^i = \frac{1}{k} \sum_{j=1}^k \bar{Q}_j^i(z_j). \quad (9)$$

To decide whether a granule must be updated or not, the concept of data sample admissibility is used. A data sample z_k is said to be admitted by a given granule prototype \mathcal{P}_k^i if it is used to update the granule prototype parameters. In this sense, two criteria are used to evaluate the data sample admissibility:

$$d(z_k, \mu_k^i) < \nu, \quad (10)$$

$$Q_k^i > Q_{k-1}^i, \quad (11)$$

where $\nu = (\chi^2)^{-1}(\gamma, n)$ is a threshold parameterized by the inverse of chi-squared statistic with $n + m$ degrees of freedom, leading EEFIG prototype to cover around $100\gamma\%$ of the stream sample. A data sample z_k which does not meet the first condition (10) for some granule is denominated an anomaly. In parallel, as the data samples are available and evaluated, a structure named tracker whose objective is to

follow the data stream dynamics to indicate change points, is established.

The tracker is parameterized by a mean vector μ_k^{tr} and an inverse covariance matrix Σ_k^{tr} , which are recursively updated (Moshtaghi, Leckie, & Bezdek, 2016). A new granule is created if the following conditions hold:

1. The tracker is c -separated from all the existing granule prototypes. The c -separation condition is expressed as follows

$$\|\mu_k^{\text{tr}} - \mu_k^i\| \geq c \sqrt{n_z \max(\bar{\xi}(\Sigma_k^{\text{tr}}), \bar{\xi}(\Sigma_k^i))}, \quad (12)$$

for all $\mathcal{G}_k^i \in \mathbb{G}_k$, where $\bar{\xi}(\Sigma_k^{\text{tr}})$ is the largest eigenvalue of Σ_k and, $c \in [0, \infty)$ specifies the separation level. Here, c is assumed as 2.

2. The number of consecutive anomalies is $n_a > \zeta$ where ζ is a hyper-parameter defined by the user to control the minimum amount of anomalies which may enable the rule creation.

3.3. EEFIG-based degradation modelling and RUL estimation

Based on the EEFIG model described in the previous section, the following Takagi-Sugeno fuzzy model is proposed for the degradation modeling

$$\begin{aligned} \text{Rule } i : & \text{IF } z_k \text{ is } \mathcal{G}_k^i \\ \text{THEN } & y_k^i = \theta_k^i \top [y_{k-1}, y_{k-2}, \dots, y_{k-L}] \top, \end{aligned} \quad (13)$$

for $i \in \mathbb{N}_{\leq C_k}$, where $y_k \in \mathbb{R}$ is the health index, $z_k \in \mathbb{R}^{n_z}$ is the vector of premise variables, $\theta_k^i \in \mathbb{R}^L$ are the coefficients of the i -th fuzzy rule at instant k , $L \in \mathbb{N}$ is the number of regressors in the autoregressive consequent, and $C_k \in \mathbb{N}$ is the number of rules at instant k . Using the center-of-gravity defuzzification for (13), the health index y_k is

$$y_k = \sum_{i=1}^{C_k} g_k^i(z_k) \theta_k^i \top [y_k \ y_{k-1} \ \dots \ y_{k-L+1}] \top \quad (14)$$

$$\Theta_k h_k(\mathbf{y}_k), \quad (15)$$

where

$$\begin{aligned} \Theta_k &= [\theta_k^1 \top \ \dots \ \theta_k^{C_k} \top], \\ \mathbf{y}_j &= \begin{bmatrix} y_j \\ \vdots \\ y_{j-L+1} \end{bmatrix}, \quad h_k(\mathbf{y}_l) = \begin{bmatrix} g_k^1(z_k) \mathbf{y}_j \\ \vdots \\ g_k^{C_k}(z_k) \mathbf{y}_j \end{bmatrix}. \end{aligned}$$

As described in (Cordovil et al., 2020; Cordovil et al., 2022), the consequent parameters Θ_k are estimated based on Re-

ursive Least Squares (RLS) methods. In particular, here we use the Sliding-windowed Fuzzily Weighted Recursive Least Squares (SFWRLS) where the weights are the membership degrees and the data window contains the last φ samples:

$$H_k = [h_k(\mathbf{y}_{k-1}) \ \dots \ h_k(\mathbf{y}_{k-\varphi})] \quad (16)$$

$$X_k = [y_k \ \dots \ y_{k-\varphi+1}] \quad (17)$$

Therefore, the recursive equation for the SFWRLS estimator are provided as follows

$$\Upsilon_k = P_k H_k (\eta I_\varphi + H_k^\top P_k H_k)^{-1} \quad (18)$$

$$P_{k+1} = \eta^{-1} (P_k - \Upsilon_k H_k^\top P_k) \quad (19)$$

$$\Theta_{k+1} = \Theta_k + (X_k - \Theta_k H_k)^\top \Upsilon_k^\top \quad (20)$$

where $P_k \in \mathbb{R}^{L C_k \times L C_k}$ is an estimate of the inverted regularised data autocorrelation matrix, $\Upsilon_k \in \mathbb{R}^{n_x}$ is the SFWRLS gain vector, and $\eta \in (0, 1]$ is the forgetting factor.

Given the estimate of the parameters of (13), the one-step ahead prediction of the degradation at instant k is computed as follows

$$\hat{y}_{k+1|k} = \sum_{i=1}^{C_k} g_k^i(z_k) \theta_k^i \top [y_k \ y_{k-1} \ \dots \ y_{k-L+1}] \top \quad (21)$$

For any $N \in \mathbb{N}$, define

$$\hat{\mathbf{y}}_{k+N|k} = \begin{cases} [y_k, y_{k-1}, \dots, y_u]^\top, & \text{if } N = 1, \\ [\hat{y}_w, \dots, \hat{y}_{k+1}, y_k, \dots, y_u]^\top, & \text{if } 1 < N < L, \\ [\hat{y}_w, \dots, \hat{y}_u]^\top, & \text{if } N \geq L, \end{cases} \quad (22)$$

where $u = k + N - L$ and $w = k + N - 1$. The N -step ahead health index prediction $\hat{y}_{k+N|k}$ is computed as follows:

$$\hat{y}_{k+N|k} = \mathbf{A}_k \hat{\mathbf{y}}_{k+N|k}, \quad (23)$$

where $\mathbf{A}_k = \sum_{i=1}^{C_k} \omega_k^i(z_k) \theta_k^i \top$.

Based on the long term prediction described in (23), the RUL can be estimated by predicting the future health state of the system given the current and past system's condition, which are provided by $\hat{\mathbf{y}}_{k+N|k}$ and z_k . Indeed, the RUL can be defined as the amount of time until the system's health index reaches a predefined threshold, that is:

$$\hat{\text{RUL}}_k = \inf \{N \in \mathbb{Z}_{\geq 0} : \hat{y}_{k+N|k} \leq \eta\}, \quad (24)$$

where $\hat{\text{RUL}}_k \in \mathbb{Z}_{\geq 0}$ denotes the RUL estimate computed at instant k given the observations of degradation state until k , and η is the end of life threshold, which must be defined based on historic data.

3.4. Uncertainty quantification

Consider a state transition function given by a Takagi-Sugeno (TS) model, with rules as in (13). The degradation propagation (23) can be rewritten as

$$\hat{y}_{k+N|k} = \mathbf{A}_k \mathbf{y}_{k+N|k} + \epsilon_{k+N}, \quad \forall N > 0. \quad (25)$$

To account for prediction uncertainties, white Gaussian noise is added to (25) from

$$\epsilon_k \sim \mathcal{N}(0, \sigma_\epsilon^2), \quad (26)$$

where σ_ϵ^2 is considered constant. The noise variance can be estimated through Monte Carlo simulations using the consequent parameters' covariance matrix estimated via RLS until time instant k (Camargos et al., 2020) or by recursively tracking the covariance of estimation errors through the on-line learning operation, i.e., for time instances $n \in \mathbb{N}_{\leq k}$ (Camargos et al., 2021). In the univariate case, the mean error is recursively tracked as

$$\Delta_{\epsilon,k} = \epsilon_k - \hat{\mu}_{\epsilon,k-1}, \quad (27)$$

$$\hat{\mu}_{\epsilon,k} = \hat{\mu}_{\epsilon,k-1} + \frac{1}{k} \Delta_{\epsilon,k}. \quad (28)$$

The initial mean error is $\hat{\mu}_{\epsilon,0} = 0$. Given the estimated mean error, the sum of squares is obtained recursively from

$$s_{\epsilon,k} = s_{\epsilon,k-1} + (\epsilon_k - \hat{\mu}_{\epsilon,k-1})^2, \quad (29)$$

being $s_{\epsilon,0} = 0$. The variance σ_ϵ^2 in (26), used for long-term prediction, is then approximated by the error covariance matrix at time instant n :

$$\sigma_\epsilon^2 = \frac{s_{\epsilon,k}}{k-1}. \quad (30)$$

3.5. Uncertainty propagation

After obtaining the initial uncertainty in one step estimates, its long term propagation considers the input vector (22) to be a vector composed of estimated random variables. Note that if $N = 1$, the previous degradation states are known and, naturally, are non-random variables. Accordingly, the output \hat{x}_{k+N} of the state transition relation (25) is also a random variable. Computing variances in a multi-step prediction framework is needed for uncertainty propagation. The first step gives

$$\begin{aligned} \text{Var}(\hat{y}_{k+1|k}) &= \mathbf{A}_k \text{Cov}(\mathbf{y}_{k+1|k}) \mathbf{A}_k^\top + \sigma_\epsilon^2 \\ &= \mathbf{A}_k \mathbf{\Lambda}_1^L \mathbf{A}_k^\top + \sigma_\epsilon^2 \\ &= \sigma_\epsilon^2 \\ &= \lambda_1^2, \end{aligned} \quad (31)$$

in which $\mathbf{\Lambda}_N^L \triangleq \text{Cov}(\mathbf{y}_{k+N|k})$, and $\lambda_N^2 \triangleq \text{Var}(\hat{y}_{k+N|k})$. Note that $\mathbf{\Lambda}_1^L = 0$, since previous degradation states are known

at $N = 1$. Then, the **N-step variance** is computed recursively as

$$\text{Var}(\hat{y}_{k+N|k}) = \mathbf{A}_k \mathbf{\Lambda}_N^L \mathbf{A}_k^\top + \sigma_\epsilon^2. \quad (32)$$

The covariance matrix of the random vector $\mathbf{y}_{k+1|k}$ is

$$\mathbf{\Lambda}_N^L = \begin{bmatrix} \lambda_{N-1}^2 & \cdots & \lambda_{N-L} \lambda_{N-1} \hat{\rho}_{L,1} \\ \vdots & \ddots & \vdots \\ \lambda_{N-1} \lambda_{N-L} \hat{\rho}_{1,L} & \cdots & \lambda_{N-L}^2 \end{bmatrix}. \quad (33)$$

Moreover, $\lambda_i^2 = 0$ when $i < 0$, meaning that x_{k+N} is known. The covariance matrix (33) is weighted by Pearson correlation coefficients, $\hat{\rho}$, estimated through historic data.

Considering the degradation to be a random variable with Gaussian distribution, whose expected value is propagated by successive iterations of (25), then RUL lower and upper bounds at an $(\alpha)(100)\%$ significance level are given as

$$\hat{\text{RUL}}_k^{\text{lb}} = \inf \{N \in \mathbb{Z}_{\geq 0} : \hat{y}_{k+N|k} + z_{1-\frac{\alpha}{2}} \lambda_N \leq \eta\}, \quad (34a)$$

$$\hat{\text{RUL}}_k^{\text{ub}} = \inf \{N \in \mathbb{Z}_{\geq 0} : \hat{y}_{k+N|k} + z_{\frac{\alpha}{2}} \lambda_N \leq \eta\}. \quad (34b)$$

4. EXPERIMENTAL SETUP

To evaluate the proposed data-driven prognostics based on evolving fuzzy degradation model, we use the accelerated aging IGBT dataset from the NASA Ames Research Center¹. This dataset contains sensor data from four devices. In particular, there are aging time series for the collector-emitter voltage (V_{CEon}), gate-emitter voltage, collector current, thermal and electrical resistance, and the times in which the switch is on and off. The health index y_k is selected to be

$$y_k = SD(\arctan(V_{\text{CEon}})), \quad (35)$$

and the premise variables vector is

$$z_k = [\bar{E}_k \ \bar{E}_{k-1} \ \bar{E}_{k-\tau+1}]^\top \quad (36)$$

where \bar{E}_k is the energy of V_{CEon} described in Table 2.

The proposed evolving fuzzy prognostics require the tuning of some hyper-parameters, namely: L , the number of lags in the autoregressive model for the health index y_k (cf. (13)); τ , the number of lags of E_k used in the premise vector z_k ; η , the forgetting factor of SFWRLS (cf. (18), (19) and (20)); φ , the size of the data windows used in SFWRLS (cf. (16) and (17)); and ζ , the number of necessary consecutive anomalies to enable the granule creation.

For choosing the hyper-parameters of the proposed algorithm, we designate a test dataset regarding one of the four devices

¹The dataset is available for download in ti.arc.nasa.gov/project/prognostics-data-repository

and perform a grid search to solve the following problem

$$\ell(D) = \arg \max_l \sum_{k=1}^{EOL_D} kRA_k(D, l) \quad \text{s.t.} \quad l \in \mathcal{L} \quad (37)$$

where $l = (L, \tau, \eta, \varphi, \zeta)$ is the vector of hyper-parameters, $\mathcal{L} = [2, 5] \times [2, 5] \times [0.96, 1] \times [2, 6] \times [2, 6]$ is the search space, EOL_D is the end of life of the D -th device, and $\ell(D)$ are the optimal parameters within the the search space \mathcal{L} , and the Relative Accuracy (RA) is

$$RA_k = 1 - \frac{|RUL_k - \hat{RUL}_k|}{RUL_k}, \quad (38)$$

5. EXPERIMENTAL RESULTS

In this section, the results for the RUL prediction for IGBTs based on evolving fuzzy models are presented and discussed. For evaluating the results, the Mean Absolute Percentage Error (MAPE) is used as figure of merit:

$$MAPE_k = \frac{100}{EOL - k + 1} \sum_{i=k+1}^{EOL} \left| \frac{RUL_i - \hat{RUL}_i}{RUL_i} \right|, \quad (39)$$

where EOL is the end of number of the UUT; r_k and \hat{EOL}_k are the current and estimated RUL at k , respectively.

Table 3 provides the MAPE results computed from $k = 20$. Notice that the EEFIG-based prognostics was able to guarantee MAPE results below of 50% for the IGBT devices 1, 2 and 4. However, the third IGBT presents more challenging data which results in higher MAPE for any parameter set.

Table 3. MAPE₂₀ results

		Parameter tuning dataset			
		1	2	3	4
UUT dataset	1	20.3560	31.5868	59.6492	48.3150
	2	23.0306	15.1883	34.4047	15.2904
	3	67.4113	76.0535	70.8395	74.9962
	4	40.7839	31.4586	37.6570	28.2187

Figures 5-7 depict the RUL prediction results in α - λ plots with accuracy cones of $\pm 30\%$. In particular, Figure 5 presents the results for the second IGBT using the parameters obtained by solving (37) for the dataset extracted from the fourth IGBT. Figure 6 presents the results for the first IGBT using the parameters obtained by solving (37) for the dataset extracted from the second IGBT. And, Figure 7 presents the results for the second IGBT using the parameters obtained by solving (37) for the dataset extracted from the third IGBT.

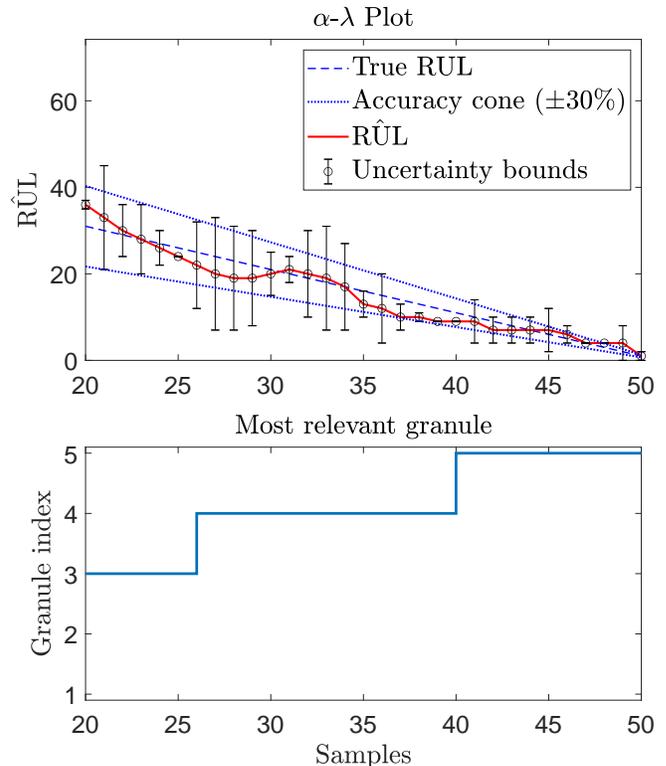


Figure 5. RUL prediction for the 2nd IGBT with parameters obtained for the test dataset with data from the 4th IGBT.

In Figures 5 and 6, notice that the RUL predictions remain inside of the accuracy cone in most of the time, and the true RUL tends to be within the predicted RUL bounds. However, the results become considerably worse for the Figure 7, as already indicated in Table 3.

One of the key advantages of applying evolving fuzzy methods is the interpretability. In this regard, the bottom plots of Figures 5, 6, and 7 indicate the granule with maximum membership degree at each sample. It is possible that news granules are being created and becoming more relevant since they are capturing novel degradation stages. Indeed, the transitions between the most relevant granules could be used as an failure or degradation stage indicator.

6. CONCLUSIONS

This paper presented a novel data-driven prognostics approach based on the evolving granular fuzzy models denominated EEFIG for IGBTs. The EEFIG is able to learn degradation processes from data-stream adapting the parameters of the degradation process representation and modifying its structure by means of granule creation for representing novel stages of the degradation process. The results indicate that the application of EEFIG for data-driven prognostics of IGBTs is promising, mainly due to its interpretability features.

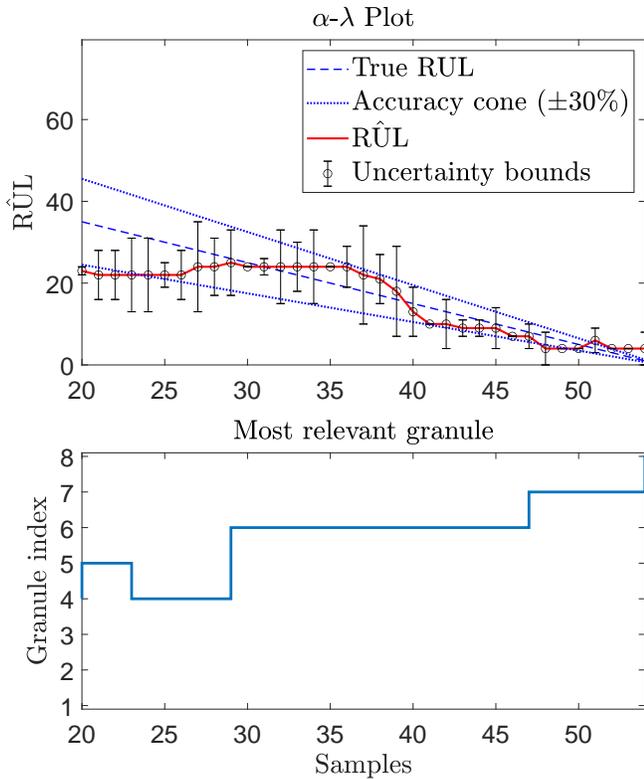


Figure 6. RUL prediction for the 1st IGBT with parameters obtained for the test dataset with data from the 2nd IGBT.

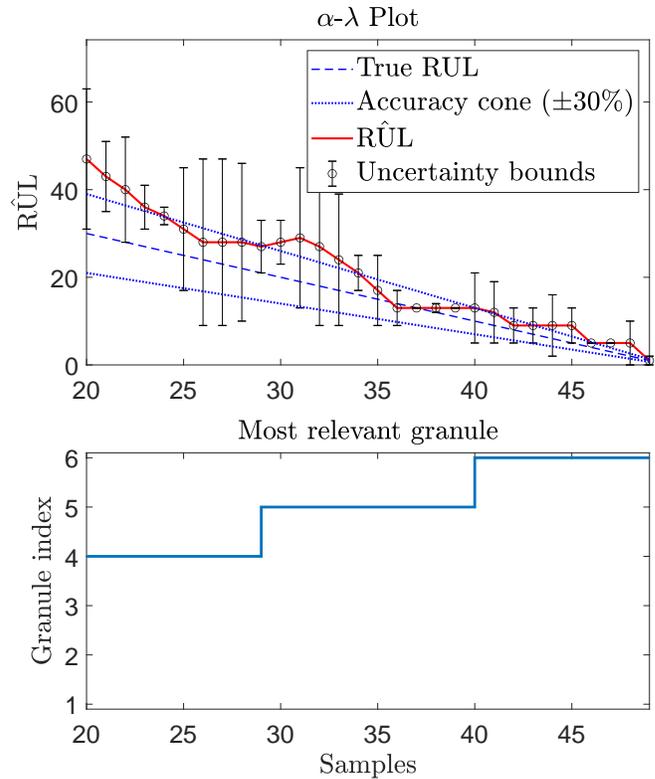


Figure 7. RUL prediction for the 2nd IGBT with parameters obtained for the test dataset with data from the 3rd IGBT.

ACKNOWLEDGMENT

This work has been co-financed by the Spanish State Research Agency (AEI) and the European Regional Development Fund (ERFD) through the project SaCoAV (ref. MINECO PID2020-114244RB-I00), by the European Regional Development Fund of the European Union in the framework of the ERDF Operational Program of Catalonia 2014-2020 (ref. 001-P-001643 Looming Factory), by the DGR of Generalitat de Catalunya (SAC group ref. 2017/SGR/482), by the Brazilian agencies CNPq, FAPEMIG, FAPEAM, and by the PROPG-CAPES/FAPEAM Scholarship Program.

REFERENCES

Ahsan, M., Stoyanov, S., & Bailey, C. (2016). Data driven prognostics for predicting remaining useful life of IGBT. In *39th International Spring Seminar on Electronics Technology (ISSE)* (p. 273-278). doi: 10.1109/ISSE.2016.7563204

Ahwiadi, M., & Wang, W. (2022). An adaptive evolving fuzzy technique for prognosis of dynamic systems. *IEEE Transactions on Fuzzy Systems*, 30(3), 841-849. doi: 10.1109/TFUZZ.2021.3049916

Alghassi, A., Perinpanayagam, S., & Samie, M. (2016). Stochastic RUL Calculation Enhanced With TDNN-

Based IGBT Failure Modeling. *IEEE Transactions on Reliability*, 65(2), 558-573. doi: 10.1109/TR.2015.2499960

Angelov, P. (2012). *Autonomous learning systems: from data streams to knowledge in real-time*. John Wiley & Sons.

Brown, D., Abbas, M., Ginart, A., Ali, I., Kalgren, P., & Vachtsevanos, G. (2010, 01). Turn-off time as a precursor for gate bipolar transistor latch-up faults in electric motor drives. *Annual Conference of the PHM Society (PHM)*.

Camargos, M., Bessa, I., D’Angelo, M. F. S. V., Cosme, L. B., & Palhares, R. M. (2020, November). Data-driven prognostics of rolling element bearings using a novel error based evolving takagi–sugeno fuzzy model. *Applied Soft Computing*, 96, 106628. doi: 10.1016/j.asoc.2020.106628

Camargos, M., Bessa, I., Junior, L. A. Q. C., Coutinho, P., Leite, D. F., & Palhares, R. M. (2021). Evolving fuzzy system applied to battery charge capacity prediction for fault prognostics. In *Atlantis studies in uncertainty modelling*. Atlantis Press. doi: 10.2991/asum.k.210827.010

Celaya, J., Saxena, A., Saha, S., & Goebel, K. (2011, 01). Prognostics of power MOSFETs under thermal stress accelerated aging using data-driven and model-based methodologies. *Annual Conference of the PHM Society*

- (PHM), 2.
- Cordovil, L. A. Q., Coutinho, P. H. S., Bessa, I., D'Angelo, M. F. S. V., & Palhares, R. M. (2020). Uncertain data modeling based on evolving ellipsoidal fuzzy information granules. *IEEE Transactions on Fuzzy Systems*, 28(10), 2427-2436. doi: 10.1109/TFUZZ.2019.2937052
- Cordovil, L. A. Q., Coutinho, P. H. S., Bessa, I., Peixoto, M. L. C., & Palhares, R. M. (2022, January). Learning event-triggered control based on evolving data-driven fuzzy granular models. *International Journal of Robust and Nonlinear Control*, 32(5), 2805–2827.
- Degrenne, N., Kawahara, C., & Mollov, S. (2019, 09). Prognostics framework for power semiconductor igt modules through monitoring of the on-state voltage. *Annual Conference of the PHM Society, 11*. doi: 10.36001/phmconf.2019.v1i1.829
- Eleffendi, M. A., & Johnson, C. M. (2016). Application of kalman filter to estimate junction temperature in igt power modules. *IEEE Transactions on Power Electronics*, 31(2), 1576-1587. doi: 10.1109/TPEL.2015.2418711
- Gouriveau, R., Medjaher, K., & Zerhouni, N. (2016). *From prognostics and health systems management to predictive maintenance 1: Monitoring and prognostics*. doi: 10.1002/9781119371052
- Hanif, A., Yu, Y., DeVoto, D., & Khan, F. (2019). A comprehensive review toward the state-of-the-art in failure and lifetime predictions of power electronic devices. *IEEE Transactions on Power Electronics*, 34(5), 4729-4746. doi: 10.1109/TPEL.2018.2860587
- Haque, M. S., Choi, S., & Baek, J. (2018). Auxiliary Particle Filtering-Based Estimation of Remaining Useful Life of IGBT. *IEEE Transactions on Industrial Electronics*, 65(3), 2693-2703. doi: 10.1109/TIE.2017.2740856
- Ismail, A., Saidi, L., Sayadi, M., & Benbouzid, M. (2019). Gaussian Process Regression Remaining Useful Lifetime Prediction of Thermally Aged Power IGBT. In *45th Conference of the IEEE Industrial Electronics Society (IECON)* (Vol. 1, p. 6004-6009). doi: 10.1109/IECON.2019.8926710
- Ismail, A., Saidi, L., Sayadi, M., & Benbouzid, M. (2020). Remaining useful life estimation for thermally aged power insulated gate bipolar transistors based on a modified maximum likelihood estimator. *International Transactions on Electrical Energy Systems*, 30(6), 1-18.
- Javed, K., Gouriveau, R., Zerhouni, N., & Nectoux, P. (2015). Enabling health monitoring approach based on vibration data for accurate prognostics. *IEEE Transactions on Industrial Electronics*, 62(1), 647-656. doi: 10.1109/TIE.2014.2327917
- Kabir, A., Bailey, C., Lu, H., & Stoyanov, S. (2012, 05). A review of data-driven prognostics in power electronics. In (Vol. 6273136, p. 189-192). doi: 10.1109/ISSE.2012.6273136
- Lu, B., & Sharma, S. K. (2009). A literature review of igt fault diagnostic and protection methods for power inverters. *IEEE Transactions on Industry Applications*, 45, 1770-1777.
- Moshtaghi, M., Leckie, C., & Bezdek, J. C. (2016, June). Online clustering of multivariate time-series. In *Proceedings of the SIAM international conference on data mining*. doi: 10.1137/1.9781611974348.41
- Nguyen, H., & Kwak, S. (2020, 12). Enhance reliability of semiconductor devices in power converters. *Electronics*, 9, 2068. doi: 10.3390/electronics9122068
- Pedrycz, W., & Wang, X. (2016). Designing fuzzy sets with the use of the parametric principle of justifiable granularity. *IEEE Transactions on Fuzzy Systems*, 24(2), 489–496.
- Saha, B., Celaya, J. R., Wysocki, P. F., & Goebel, K. F. (2009). Towards prognostics for electronics components. In *2009 IEEE Aerospace Conference* (p. 1-7). doi: 10.1109/AERO.2009.4839676
- Sonnenfeld, G., Goebel, K., & Celaya, J. R. (2008). An agile accelerated aging, characterization and scenario simulation system for gate controlled power transistors. In *2008 IEEE Autotestcon* (p. 208-215). doi: 10.1109/AUTEST.2008.4662613
- Wang, G., Shi, P., Wang, B., & Zhang, J. (2014). Fuzzy *n*-ellipsoid numbers and representations of uncertain multichannel digital information. *IEEE Transactions on Fuzzy Systems*, 22(5), 1113–1126.
- Yang, S., Bryant, A., Mawby, P., Xiang, D., Ran, L., & Tavner, P. (2011). An industry-based survey of reliability in power electronic converters. *IEEE Transactions on Industry Applications*, 47(3), 1441-1451. doi: 10.1109/TIA.2011.2124436
- Zhang, Y., Liu, Y., Li, C., & Li, J. (2020, November). Analysis of fault precursor parameters under accelerated aging tests for IGBT modules. In *17th China International Forum on Solid State Lighting & International Forum on Wide Bandgap Semiconductors China*. doi: 10.1109/sslchinaifws51786.2020.9308699

State of Health and Lifetime Prediction of Lithium-ion Batteries Using Self-learning Incremental Models

Murilo Camargos^{1,2,3} and Plamen Angelov^{1,2,3}

¹ *School of Computing and Communications, Lancaster University, Lancaster, LA1 4WA, UK*
m.camargos@lancaster.ac.uk
p.angelov@lancaster.ac.uk

² *Lancaster Intelligent, Robotic and Autonomous Systems (LIRA) Research Centre, Lancaster, UK*

³ *The Faraday Institution, Quad One, Becquerel Avenue, Harwell Campus, Didcot, OX11 0RA, UK*

ABSTRACT

Lithium-ion batteries are key energy storage elements in the context of environmental-aware energy systems representing a crucial technology to achieve the goal of zero carbon emission. Therefore, its conditions must be monitored to guarantee the safe and reliable operation of the systems that use these components. Furthermore, lithium-ion batteries' prognostics and health management policies must cope with the nonlinear and time-varying nature of the complex electrochemical dynamics of battery degradation. This paper proposes an incremental-learning-based algorithm to estimate the State of Health (SoH) and the Remaining Useful Life (RUL) of lithium-ion batteries based on measurement data streams. For this purpose, a two-layer framework is proposed based on incremental modeling of the SoH. In the first layer, a set of representative features are extracted from voltage and current data of partial charging and discharging cycles; these features are then used to train the proposed model in a recursive procedure to estimate the battery's SoH. The second layer uses the capacity data for incremental learning of an Autoregressive (AR) model for the SoH, which will be used to propagate the battery's degradation through time to make the RUL prediction. The proposed method was applied to two datasets for experimental evaluation, one from CALCE and another from NASA. The proposed framework was able to estimate the SoH of 8 different lithium-ion cells with an average percentage error below 1.5% for all scenarios, while the lifetime model predicted the cell's RUL with a maximum average error of 25%.

Murilo Camargos et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ACRONYMS

SoH	State of Health
RUL	Remaining Useful Life
TS	Takagi-Sugeno
MF	Membership Function
RLS	Recursive Least Squares
MAPE	Mean Absolute Percentage Error
RMSPE	Root Mean Squared Percentage Error
FT	Fault Threshold
AR	Autoregressive
IC	Incremental Capacity
DV	Differential Voltage

1. INTRODUCTION

The lithium-ion batteries are key energy storage elements in the context of environmental-aware energy systems due to their recognized performance and energy density. For this reason they have been widely applied in microgrids, consumer electronics, and electric vehicles. However, the safety and operation costs of those systems become dependent on the battery's storage capacity and lifetime. Indeed, the battery's charge capacity is progressively reduced due to the aging and repeated charging (discharging) cycles (Birkl, Roberts, McTurk, Bruce, & Howey, 2017). Therefore, the PHM of lithium-ion batteries is essential to improve the reliability of those systems and extend their lifetime. In this sense, the PHM methodologies for lithium-ion batteries (Y. Zhang & Li, 2022; Omariba, Zhang, & Sun, 2018; Ge, Liu, Jiang, & Liu, 2021) are responsible for estimating their State of Health (SoH) and Remaining Useful Life (RUL).

In particular, the SoH of a battery denotes the ratio of its current parameters (e.g., charge capacity, impedance, and power)

and the same parameters at the beginning of its life (Cai, Lin, & Liao, 2022). Otherwise, the RUL of a battery is defined as time between the current observation instant and the battery collapse instant, denominated the battery's end-of-life (Dong, Han, & Wang, 2021). For a battery approach, it is necessary to estimate the SoH and RUL, since these variables cannot be directly measured in an everyday battery use. Some techniques as Coulomb counting and peak tracking in Incremental Capacity (IC)/Differential Voltage (DV) curves often require long measurement duration, are prone to noise amplification and might not be well generalized for different cells (Richardson, Birkel, Osborne, & Howey, 2019).

The physics of the battery's degradation can be used to derive mathematical models, e.g., battery equivalent circuit models or electrochemical models, to enable the SoH estimation and RUL prediction. However, those model-based methodologies (Downey, Lui, Hu, Laflamme, & Hu, 2019; Lui et al., 2021) required high fidelity models for the degradation process that is usually nonlinear and time-varying for batteries. Since some of those models' parameters are not known *a priori*, the application of model-based methods might also require parameter estimation techniques which becomes more challenging for complex mathematical models.

Hence, the use of data-driven methodologies (Wu, Fu, & Guan, 2016) is specially attractive in applications to SoH and RUL estimation of lithium-ion batteries. In general, the data-driven methodologies for lithium-ion batteries can be classified into three groups:

1. **Empirical methods** are built based on some historical data used to estimate parameters for a chosen structure, e.g., exponential models (Cai et al., 2022), autoregressive models (M. Camargos et al., 2021), and neural networks (Q. Zhang et al., 2022).
2. **Stochastic methods** describe degradation phenomenons as stochastic processes and the SoH as a random variable. The parameters of those stochastic models can be estimated through different methodologies, such as Gaussian process regression (Richardson et al., 2019; X. Li, Wang, & Yan, 2019) and Bayesian filtering (Si, 2015).
3. **Signal-based methods** aims at obtaining the relation between the capacity loss and the measured signals properties which can be extracted in the time or frequency domain (Khaleghi, Firouz, Mierlo, & den Bossche, 2019; Wang, Pan, Liu, Cheng, & Zhao, 2016).

Most of the aforementioned data-driven methodologies establish stiff relations between the process data and the degradation processes related to the SoH and RUL estimation. However, the battery's degradation and its relation with the measurable data is complex, time-varying, and tends to be particular for each cell under test. This context motivates the development of methodologies which are able to adapt themselves to the cell's behavior without compromising the gen-

erality ability. In this sense, some adaptive or incremental learning methodologies have recently been proposed for battery's SoH and RUL estimation (J. Zhang et al., 2022; Qin, Zhao, & Liu, 2022; Si, 2015). Although those methods are able to update their parameters to provide a better representation of the degradation processes based on the data streams, they are unable to modify their complexity to capture novel dynamic behaviors.

In this regard, evolving systems are effective tools for obtaining incremental models which update their structure and adapt their parameters through autonomous learning from data streams (Angelov, 2012). For this reason, evolving systems have been effectively applied for dealing with complex and time-varying dynamics aiding to solve different problems, such as fault diagnosis (Shah & Wang, 2021), classification (Soares, Angelov, & Gu, 2020), time-series prediction and forecasting (Severiano, de Lima e Silva, Cohen, & Guimarães, 2021), system identification (Škrjanc, 2021), and learning-based control (Cordovil, Coutinho, Bessa, Peixoto, & Palhares, 2022). Recently, the use of evolving fuzzy models has been proposed for solving the RUL prediction problems (M. O. Camargos, Bessa, D'Angelo, Cosme, & Palhares, 2020), including with applications to lithium-ion batteries (Ahwiadi & Wang, 2022; M. Camargos et al., 2021). In addition to flexibility and adaptability of those incremental models, the evolving fuzzy models provides interpretability for the data-driven approaches. For example, the prognostics based on evolving fuzzy models allows to relate the increment of the model structure to the degradation stage.

This paper addresses the problem of data-stream based SoH estimation and RUL prediction for lithium-ion batteries. To solve this problem, it is proposed a two-layer framework based on incremental modeling of the SoH. The first layer extracts features related to the voltage from charging cycles, then, uses the extracted features for incremental learning of the SoH behavior and SoH estimation. The second layer uses the capacity data for incremental learning of an Autoregressive (AR) model for the SoH, which is then applied for RUL prediction.

The remainder of this paper is organized as follows: Section 2 provides an overview on the class of self-learning incremental models used in this paper; Section 3 describes the proposed methodology for SoH estimation and RUL prediction; Section 4 presents the experimental procedures and setup; Section 5 presents the results for SoH estimation and RUL prediction applied to two lithium-ion batteries' datasets; and Section 6 draws the conclusions and indicates further research directions.

2. SELF-LEARNING INCREMENTAL MODELS

The self-learning incremental models used in this paper are represented using the Takagi-Sugeno (TS) representation. This

type of representation can be seen as a mixture of linear models whose mixing probabilities are given by fuzzy relations. Moreover, it is a powerful modeling technique that is capable of approximating nonlinear dynamics, multiple operating modes and significant parameter and structure variations (Angelov & Filev, 2004).

The TS fuzzy models are rule-based models composed by C IF-THEN rules that are used as an inference system. The antecedents are represented by fuzzy relations between the input data and a knowledge base of fuzzy sets while the consequents are usually linear functions (Nguyen et al., 2019). Each rule is represented as:

$$\text{Rule } i: \mathbf{IF} (x_1 \text{ is } \Phi_{i1}) \mathbf{AND} \dots \mathbf{AND} (x_{n_x} \text{ is } \Phi_{in_x}) \quad (1) \\ \mathbf{THEN} \hat{y}_i = \mathbf{a}_i^\top \tilde{\mathbf{x}}$$

where $\mathbf{x} = [x_1, \dots, x_{n_x}]^\top \in \mathbb{R}^{n_x}$ is the vector of premise variables, $\mathbf{a}_i \in \mathbb{R}^{n_x+1}$ is the vector of estimated consequent parameters, and $\tilde{\mathbf{x}} = [1 \quad \mathbf{x}^\top]^\top$. Moreover, $(x_j \text{ is } \Phi_{ij})$ denotes the fuzzy relation between x_j and the fuzzy set Φ_{ij} for $i \in \mathbb{N}_{\leq C}$ and $j \in \mathbb{N}_{\leq n_x}$. Throughout the text, $\mathbb{N}_{\leq k}$ will be used to denote the set of natural numbers up to k , such that $\mathbb{N}_{\leq k} = \{1, 2, \dots, k\}$.

The fuzzy relation in the antecedents will define the activation degree of each rule, i.e., the level of contribution of each local linear model to the overall output. The activation degree is given by a Membership Function (MF) $\varphi_{ij}: \mathbb{R} \rightarrow [0, 1]$ that maps a given input's component x_j to the unit partition, for $j \in \mathbb{N}_{\leq n_x}$. For Gaussian-like antecedent fuzzy sets, the MF is of the form:

$$\varphi_{ij}(x_j) = \exp(-\alpha \|x_j - x_{ij}^*\|^2), \quad (2)$$

where x_{ij}^* is the j -th component of the focal point of the i -th rule, $\alpha = 4/r^2$ and r defines the radius of the neighborhood of a data point, also known as the model's zone of influence. According to (Angelov & Filev, 2004), too large a value of r leads to averaging while too small values leads to over-fitting. In general, values of $r \in [0.3, 0.5]$ can be recommended (Angelov & Filev, 2004; Chiu, 1994). The final activation degree of the i -th rule is defined as the Cartesian product or conjunction of respective fuzzy sets:

$$w_i(\mathbf{x}) = \bigcap_{j=1}^{n_x} (x_j \text{ is } \Phi_{ij}) = \prod_{j=1}^{n_x} \varphi_{ij}(x_j). \quad (3)$$

The output of the TS fuzzy model is a convex combination among C consequent linear models weighted by the rules' activation degrees. The activation degrees must comply with the convex sum property, i.e., they need to be non-negative

and sum one. From the center average defuzzification, the overall model output is given as

$$\hat{y} = \sum_{i=1}^C h_i(\mathbf{x}) \mathbf{a}_i^\top \tilde{\mathbf{x}} \quad (4)$$

in which

$$h_i(\mathbf{x}) = \frac{w_i(\mathbf{x})}{\sum_{m=1}^C w_m(\mathbf{x})}. \quad (5)$$

Given a set of input and output data, the problem of identifying the TS model, i.e., finding the number of rules, the focal points in Eq. (2) and the parameters of the linear subsystems in Eq. (4), is divided into two parts:

1. Finding the antecedents' focal points: $\{\mathbf{x}_1^*, \dots, \mathbf{x}_C^*\}$
2. Finding the parameters of each linear subsystem: $\{\mathbf{a}_1, \dots, \mathbf{a}_C\}$

This first task can be solved by clustering the input-output data space while the second task can be solved by computing each linear model's parameters in the least-squares sense. In (Angelov & Filev, 2004), online learning strategies for these tasks are given. In such cases, the number of clusters changes as new data samples becomes available, therefore, C becomes C_k and both the antecedents and consequents parameters also change in time, becoming $\{\mathbf{x}_{k1}^*, \dots, \mathbf{x}_{kC_k}^*\}$ and $\{\mathbf{a}_{k1}, \dots, \mathbf{a}_{kC_k}\}$.

In the clustering problem, a recursive variation of the so-called subtractive clustering (Chiu, 1994) algorithm is given. The proposed algorithm uses a Cauchy type function of first order to represent the potential of each data point to become a focal point. This function enables recursive calculation and is both monotonic and inversely proportional to the distance between two data points. The computed potential is then used to decide whether the new data point will be used to replace an old focal point or will represent a new focal point, or cluster center. The consequent parameters are updated using the Recursive Least Squares (RLS) algorithm.

The rule-base model will dynamically upgrade the number of clusters in the input-output data space or modify existing ones, while preserving rules that represents old knowledge. The details of this procedure can be found in (Angelov & Filev, 2004).

3. STATE OF HEALTH AND LIFETIME PREDICTION

In order to estimate the SoH and to predict the lifetime, i.e., estimate its RUL, of the batteries, we propose the a parallel architecture using two self-learning incremental models as described in Section 2. As shown in Figure 1, one model will be used to estimate the SoH while another one will be used to predict the RUL. They use the same learning procedure, as

described by (Angelov & Filev, 2004); however, their inputs are different: the SoH predictor uses extracted features from partial charge procedures while the RUL predictor uses past values of the charge capacity time-series to predict the next SoH, i.e., it is an AR model.

The parameter set for each model is given as:

$$\theta_k^p = \{C_k^p, \mathbf{x}_{k1}^{p*}, \dots, \mathbf{x}_{kC_k^p}^{p*}, \mathbf{a}_{k1}^p, \dots, \mathbf{a}_{kC_k^p}^p\} \quad (6)$$

where the superscript $p \in [1, 2]$ indicates the SoH model and the lifetime model respectively. The estimation on both models is done as:

$$\hat{y}_k^p = f_p(\mathbf{x}_k^p | \theta_{k-1}^p, \Omega_p) \quad (7)$$

where $f_p: \mathbb{R}^{n_x^p} \rightarrow \mathbb{R}$ is a TS fuzzy model in the form of Eq. (4), n_x^p is the dimension of input \mathbf{x}_k^p , and Ω_p is a set of time-invariant parameters. The learning procedure to update the parameter set is given as

$$\theta_{k|k-1}^p = h(\theta_{k-1}^p, \mathbf{x}_k^p, y_k, \hat{y}_k^p, \Omega_p). \quad (8)$$

Both models use the same learning procedure h . The time-invariant parameter set of the SoH model contains only the model's zone of influence $\Omega_1 = \{r_1\}$ while the lifetime model also contains the number of past capacity values to predict the next one $\Omega_2 = \{r_2, L\}$.

Their inputs are different from each other; the lifetime model takes a lagged vector as input, i.e.,

$$\mathbf{x}_k^2 = [y_{k-1}, \dots, y_{k-L}]^\top \in \mathbb{R}^{n_x^2}, \quad (9)$$

where $n_x^2 = L$ is the number of past values the lifetime model will take as inputs to predict the next one. When the prognostics task starts, i.e., when $k = t_p$, the SoH values estimated by the lifetime model will replace true SoH values in the input vector shown in Eq. (9). Then, at cycles $k \geq t_p + L$, all components in Eq. (9) will be previous estimates.

The SoH model uses features extracted from partial charge data as described in (Richardson et al., 2019). To overcome the necessity of having to identify the parameters of highly accurate battery models or the requirement of having long measurements that ensures the coverage of IC/DV curve's peaks and dealing with the noise that comes out of this process, the proposed feature extraction uses direct voltage data from partial charging procedures. After defining a specific voltage window $[V_{lb}, V_{ub}]$, M equispaced voltages are taken and the time it takes to go from one voltage to another is defined as the feature for the SoH estimation. The extracted features for the SoH estimator are given as:

$$\mathbf{x}_k^1 = [\tau_k(v_0, v_1), \dots, \tau_k(v_{M-1}, v_M)]^\top \in \mathbb{R}^{n_x^1}, \quad (10)$$

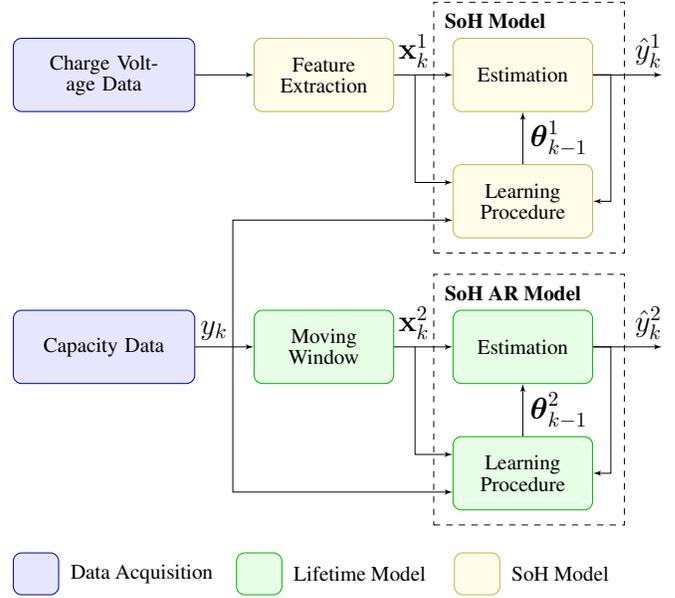


Figure 1. Parallel architecture of self incremental models for State of Health and lifetime prediction.

where $n_x^1 = M$ and

$$v_i = V_{lb} + i \cdot \frac{V_{ub} - V_{lb}}{M - 1} \quad (11)$$

in which $\tau_k(v_a, v_b)$ computes the time it takes to go from voltage v_a to voltage v_b in the k -th charging cycle.

4. EXPERIMENTAL SETUP

In order to test the proposed architecture shown in Figure 1, datasets that represent the degradation of lithium-ion batteries from two sources were used. This type of battery is commonly found in industry and commercially, e.g., in electric vehicles, microgrids, and electronic devices (X. Li et al., 2019; Saha & Goebel, 2009).

In the first dataset, the cycle aging experiments of four lithium-ion batteries (B0005, B0006, B0007, B0018) are provided by NASA Ames Prognostics Center of Excellence (PCoE)¹ (Saha & Goebel, 2007). The testbed comprises commercial lithium-ion 18650-sized rechargeable batteries from the Idaho National Laboratory; a programmable 4-channel DC electronic load and power supply; voltmeters, ammeters, and a thermocouple sensor suite; custom electrochemical impedance spectrometry equipment; and environmental chamber to impose different operational conditions. The batteries run at room temperature (23° C). Charging is done in constant mode at 1.5 A, until the voltage reaches 4.2 V. Discharging is performed at a constant current level of 2 A, until the battery voltage reaches 2.7 V (Saha & Goebel, 2009).

¹<http://ti.arc.nasa.gov/project/prognostic-data-repository>

The second dataset also contains four cycle aging experiments (CS2 35, CS2 36, CS2 37, CS2 38) of prismatic cells with graphite anode and a lithium cobalt oxide cathode. The data is provided by the Center for Advanced Life Cycle Engineering (CALCE)² from the University of Maryland. The cycling of the batteries was accomplished by multiple full charge-discharge tests using an Arbin BT2000 battery testing system under room temperature. The batteries were cycled at constant current of 1 C (1.1 A) with charging and discharging being cut off at the manufacturer’s specified cutoff voltage (from 2.7 V to 4.2 V). The capacity of the tested batteries was estimated using the Coulomb counting method (He, Williard, Osterman, & Pecht, 2011; Xing, Ma, Tsui, & Pecht, 2013).

As the cells ages, its maximum available capacity will decrease. In this paper, the SoH is defined as the relative capacity of each cell, i.e., the computed capacity at cycle k divided by the cell’s nominal capacity (2 A for NASA cells and 1.1 A for CALCE cells).

The results are obtained after defining the hyperparameters for both parallel models, i.e., Ω_1 and Ω_2 . For the SoH model, we use the standard value of $r_1 = 0.3$. For the lifetime model we perform a cross validation task to find the best values for (r_2, L) . We choose a training cell and a validation cell to perform a grid search over the parameters. The optimal parameters are given as:

$$\begin{aligned} \arg \min_{\rho, \ell} \quad & \frac{1}{H - t_P} \sum_{k=t_P}^{H-1} \frac{|r_k(\rho, \ell) - \hat{r}_k(\rho, \ell)|}{r_k(\rho, \ell)} \\ \text{subject to} \quad & \rho \in [0.3, 0.35, 0.4, 0.45, 0.5], \\ & \ell \in [1, 2, 3, 4, 5, 6, 7, 8] \end{aligned} \quad (12)$$

where $\hat{r}_k(\rho, \ell)$ is the estimated RUL at the k -th cycle of the validation cell using a model trained with the training cell, $r_k(\rho, \ell)$ is the true RUL under the same conditions, t_P is the cycle in which prognostics task starts and H is the validation cell lifespan.

Here, we define the RUL as the time elapsed between the prognostics task starting time (t_P) and the time in which the system’s degradation state reaches a given Fault Threshold (FT) (N. Li, Lei, Lin, & Ding, 2015). Formally, we can express the RUL as:

$$\hat{r}_{t_P} = \inf \left\{ n \in \mathbb{N} \mid \hat{y}_{t_P+n}^2 \geq \eta \right\}, \quad (13)$$

where \hat{r}_k denotes the RUL computed at instant t_P , given that the true values of the state of health are known up until t_P , \mathbb{N} is the natural numbers set, and η is the predefined FT. Moreover, Eq. (13) is a simplified version of the canonical RUL

²<https://web.calce.umd.edu/batteries/data.htm>

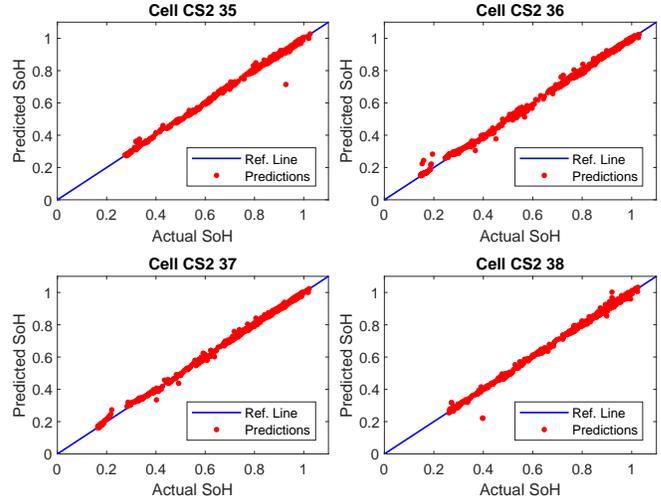


Figure 2. Predicted and actual SoH for CALCE cells.

definition by (Chiachío, Chiachío, Sankararaman, Saxena, & Goebel, 2015), where the concept of failure domain is defined.

5. RESULTS

The results section is divided into SoH estimation and lifetime prediction experiments.

5.1. SoH estimation

The feature extraction was done in a similar way of (Richardson et al., 2019). We chose the lower and upper bound voltages as roughly 75% to 100% of the charging voltage span to represent a more realistic use case scenario as full-cycle charging are not always available. Specifically, we set $V_{lb} = 3.85$ and $V_{ub} = 4.2$ with $M = 4$ equispaced voltages. Therefore, the SoH model inputs shown in Eq. (10) is defined as:

$$\mathbf{x}_k^1 = \begin{bmatrix} \tau_k(3.8500, 3.9375) \\ \tau_k(3.9375, 4.0250) \\ \tau_k(4.0250, 4.1125) \\ \tau_k(4.1125, 4.2000) \end{bmatrix}. \quad (14)$$

In this phase, no previous training was done and the SoH model learned online, as new data became available, how to predict the SoH from the inputs in Eq. (14). We set the model’s fine tuning parameter to $r_1 = 0.3$. The predicted versus the actual SoH for both CALCE and NASA datasets are shown in Figure 2 and Figure 3. In the CALCE dataset the prediction far away from the reference line are outliers in the charge/discharge measurements from the Arbin testing system.

The SoH model predictions are also evaluated according to two metrics, namely Mean Absolute Percentage Error (MAPE) and Root Mean Squared Percentage Error (RMSPE), defined

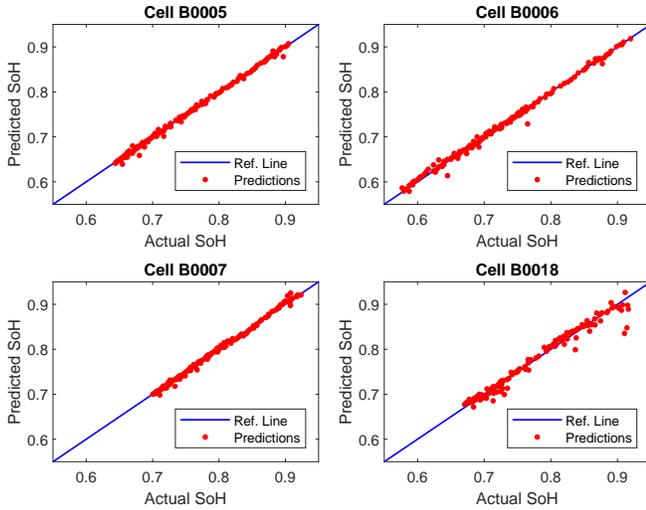


Figure 3. Predicted and actual SoH for NASA cells.

Table 1. MAPE and RMSPE values for the SoH model's predictions.

Cell ID	MAPE	RMSPE
CS2 35	0.7698%	1.5965%
CS2 36	1.2358%	3.4065%
CS2 37	0.9417%	1.8461%
CS2 38	0.8633%	2.3728%
B0005	0.3915%	0.6053%
B0006	0.5626%	0.9134%
B0007	0.3793%	0.5355%
B0018	0.9777%	1.5378%
Avg.	0.7652%	1.6017%

as:

$$\text{MAPE}(\mathbf{y}, \hat{\mathbf{y}}^1) = \frac{100}{N_T} \sum_{k=1}^{N_T} \left| \frac{y_k - \hat{y}_k^1}{y_k} \right| \quad (15)$$

$$\text{RMSPE}(\mathbf{y}, \hat{\mathbf{y}}^1) = \sqrt{\frac{1}{N_T} \sum_{k=1}^{N_T} \left(\frac{y_k - \hat{y}_k^1}{y_k} \right)^2} \quad (16)$$

where, \mathbf{y} is the vector of true SoH values, $\hat{\mathbf{y}}^1$ is the vector of prediction made by the SoH model in Figure 1 and N_T is of samples. The results in Table 1 shows that the proposed model achieved a MAPE value of less than 1.5% for all tested cells with an average of 0.77%. These results indicate the competitiveness of the SoH model in comparison to more complex models reported in the literature.

5.2. Lifetime prediction

In the lifetime prediction task, we first run a cross validation procedure to find reasonable values of the parameters Ω_2 . In order to solve the optimization problem in Eq. 12 we need to choose training and validation cells for both NASA and

Table 2. Successive lifetime prediction over multiple starting points for CALCE dataset.

k	CS2 37			CS2 38		
	r_k	\hat{r}_k	APE_k	r_k	\hat{r}_k	APE_k
100	542	394	27.31%	598	500	16.39%
150	492	418	15.04%	548	375	31.57%
200	442	407	7.92%	498	376	24.50%
250	392	350	10.71%	448	340	24.11%
300	342	294	14.04%	398	298	25.13%
350	292	230	21.23%	348	250	28.16%
400	242	197	18.60%	298	225	24.50%
450	192	200	4.17%	248	220	11.29%
500	142	170	19.72%	198	209	5.56%
550	92	84	8.70%	148	129	12.84%
600	42	32	23.81%	98	84	14.29%
Avg.			15.57%			19.85%

CALCE datasets. In this paper, this choice is done arbitrarily. For the NASA cells, we chose cells B0006 and B0018 as training and validation cells, respectively; for the CALCE cells, we chose cells CS2 35 and CS2 36 as training and validation cells, respectively.

Moreover, we define the fault thresholds of NASA and CALCE cells as $\eta_{\text{NASA}} = 80\%$ and $\eta_{\text{CALCE}} = 70\%$, respectively, due to larger lifespan of CALCE cells. The cross validation procedure yielded the following parameters: $\Omega_2^{\text{CALCE}} = \{r_2, L\} = \{0.3, 7\}$ and $\Omega_2^{\text{NASA}} = \{r_2, L\} = \{0.45, 5\}$ for CALCE and NASA, respectively.

The lifetime prediction results for different prognostics starting time (t_P) are shown in Table 2 and Table 3. For each tested cell, there is the true RUL column (r_k), the value estimated by the lifetime model (\hat{r}_k) and the absolute percentage error (APE)³. Overall, the lifetime prediction errors for all cells did not exceed 25% and the results near the actual end of life of the tested cells do not always decrease, as is expected.

Another way to depict these results is through the $\alpha - \lambda$ plot, which is used to evaluate prognostics strategies since it shows whether the predicted RUL falls within a goal region around the true RUL given by $\pm(\alpha)(100)\%$ (Lall, Lowe, & Goebel, 2012). The $\alpha - \lambda$ plots for all tested cells are shown in Figure 4 using a goal region of $\alpha = 20\%$ to define the accuracy cone. Although the prediction error exceeds the threshold of 20%, it falls below the accuracy cone in almost all the times. This happens when the algorithm underestimates the actual Remaining Useful Life, which is a situation more tolerable than when it overestimates, due to safety reasons (Nectoux et al., 2012).

³This is a version of Eq. (15) without the averaging.

Table 3. Successive lifetime prediction over multiple starting points for NASA dataset.

k	B0005			B0007		
	r_k	\hat{r}_k	APE_k	r_k	\hat{r}_k	APE_k
20	19	21	10.53%	30	24	20.00%
22	17	19	11.76%	28	22	21.43%
24	15	17	13.33%	26	20	23.08%
26	13	14	7.69%	24	18	25.00%
28	11	12	9.09%	22	16	27.27%
30	9	10	11.11%	20	15	25.00%
32	7	7	0.00%	18	13	27.78%
34	5	5	0.00%	16	12	25.00%
36	3	3	0.00%	14	10	28.57%
38	1	1	0.00%	12	9	25.00%
Avg.			6.35%			24.81%

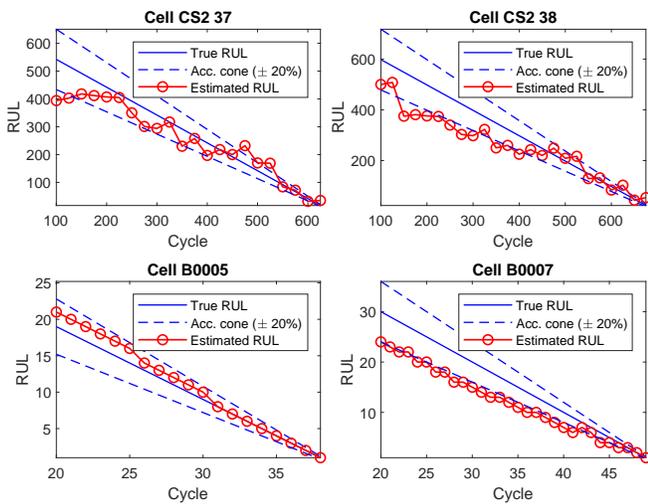


Figure 4. $\alpha - \lambda$ plot of the estimated RUL of all testing cells with goal region of $\alpha = 20\%$.

6. CONCLUSIONS

The proposed parallel architecture, composed of two self-learning incremental models, e.g., evolving TS, was capable of estimating both the SoH and the lifetime of different cells from two datasets. Moreover, the use of partial measurements of the charging voltage in the SoH model approximates us from more realistic use case scenarios, such as in electric vehicles where the charging cycles are rarely complete. Furthermore, the non-stationary behaviour seen in charge/discharge cycles can be accommodated through the operational ability to quickly change the model’s parameters and structure as new data becomes available.

The technique reached an average percentage error below 1.5% in all tested cells, indicating its competitiveness concerning other models reported in the literature that are more complex and whose training phase happens offline. Self-learning incremental models are promising methods to deal with non-

linear problems in non-stationary environments. Their structures are flexible, and their parameters can be updated recursively according to data stream changes.

Furthermore, the lifetime model managed to estimate the RUL for different cells with a low computational cost. Structural learning from scratch, quick recursive updates, and historical-data storage avoidance make self-learning incremental models quite suitable to be used in real-time prognostics systems. However, the results near the actual end of life of the tested cells do not always decrease, as is expected, indicating the proposed methodology can be improved to provide better long-term predictions. The proposed model have offered online condition monitoring and a way of fusing multivariate data streams describing the multiple-stage battery-degradation phenomenon.

ACKNOWLEDGMENT

This work was carried out with funding from the Faraday Institution (faraday.ac.uk; EP/S003053/1), grant number FIRG025.

REFERENCES

Ahwiadi, M., & Wang, W. (2022). An adaptive evolving fuzzy technique for prognosis of dynamic systems. *IEEE Transactions on Fuzzy Systems*, 30(3), 841-849. doi: 10.1109/TFUZZ.2021.3049916

Angelov, P. (2012). *Autonomous learning systems: from data streams to knowledge in real-time*. John Wiley & Sons.

Angelov, P., & Filev, D. (2004, February). An approach to online identification of takagi-sugeno fuzzy models. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 34(1), 484-498. doi: 10.1109/tsmcb.2003.817053

Birkel, C. R., Roberts, M. R., McTurk, E., Bruce, P. G., & Howey, D. A. (2017, February). Degradation diagnostics for lithium ion cells. *Journal of Power Sources*, 341, 373-386. doi: 10.1016/j.jpowsour.2016.12.011

Cai, L., Lin, J., & Liao, X. (2022, July). A data-driven method for state of health prediction of lithium-ion batteries in a unified framework. *Journal of Energy Storage*, 51, 104371. doi: 10.1016/j.est.2022.104371

Camargos, M., Bessa, I., Junior, L. A. Q. C., Coutinho, P., Leite, D. F., & Palhares, R. M. (2021). Evolving fuzzy system applied to battery charge capacity prediction for fault prognostics. In *Atlantis studies in uncertainty modelling*. Atlantis Press. doi: 10.2991/asum.k.210827.010

Camargos, M. O., Bessa, I., D’Angelo, M. F. S. V., Cosme, L. B., & Palhares, R. M. (2020, November). Data-driven prognostics of rolling element bearings using a novel error based evolving takagi-sugeno fuzzy model. *Applied Soft Computing*, 96, 106628. doi: 10.1016/j.asoc.2020.106628

- Chiachío, J., Chiachío, M., Sankararaman, S., Saxena, A., & Goebel, K. (2015). Condition-based prediction of time-dependent reliability in composites. *Reliab. Eng. Syst. Saf.*, *142*, 134–147. doi: 10.1016/j.ress.2015.04.018
- Chiu, S. L. (1994). Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, *2*(3), 267–278. doi: 10.3233/ifs-1994-2306
- Cordovil, L. A. Q., Coutinho, P. H. S., Bessa, I., Peixoto, M. L. C., & Palhares, R. M. (2022, January). Learning event-triggered control based on evolving data-driven fuzzy granular models. *International Journal of Robust and Nonlinear Control*, *32*(5), 2805–2827.
- Dong, G., Han, W., & Wang, Y. (2021, November). Dynamic bayesian network-based lithium-ion battery health prognosis for electric vehicles. *IEEE Transactions on Industrial Electronics*, *68*(11), 10949–10958. doi: 10.1109/tie.2020.3034855
- Downey, A., Lui, Y.-H., Hu, C., Laflamme, S., & Hu, S. (2019, February). Physics-based prognostics of lithium-ion battery using non-linear least squares with dynamic bounds. *Reliability Engineering & System Safety*, *182*, 1–12. doi: 10.1016/j.ress.2018.09.018
- Ge, M.-F., Liu, Y., Jiang, X., & Liu, J. (2021, April). A review on state of health estimations and remaining useful life prognostics of lithium-ion batteries. *Measurement*, *174*, 109057. doi: 10.1016/j.measurement.2021.109057
- He, W., Williard, N., Osterman, M., & Pecht, M. (2011, December). Prognostics of lithium-ion batteries based on dempster-shafer theory and the bayesian monte carlo method. *Journal of Power Sources*, *196*(23), 10314–10321. doi: 10.1016/j.jpowsour.2011.08.040
- Khaleghi, S., Firouz, Y., Mierlo, J. V., & den Bossche, P. V. (2019, December). Developing a real-time data-driven battery health diagnosis method, using time and frequency domain condition indicators. *Applied Energy*, *255*, 113813. doi: 10.1016/j.apenergy.2019.113813
- Lall, P., Lowe, R., & Goebel, K. (2012, November). Prognostics health management of electronic systems under mechanical shock and vibration using kalman filter models and metrics. *IEEE Transactions on Industrial Electronics*, *59*(11), 4301–4314. doi: 10.1109/tie.2012.2183834
- Li, N., Lei, Y., Lin, J., & Ding, S. X. (2015). An Improved Exponential Model for Predicting Remaining Useful Life of Rolling Element Bearings. *IEEE Trans. Ind. Electron.*, *62*(12), 7762–7773. doi: 10.1109/TIE.2015.2455055
- Li, X., Wang, Z., & Yan, J. (2019). Prognostic health condition for lithium battery using the partial incremental capacity and Gaussian process regression. *J. Power Sources*, *421*(February), 56–67. doi: 10.1016/j.jpowsour.2019.03.008
- Lui, Y. H., Li, M., Downey, A., Shen, S., Nemanji, V. P., Ye, H., ... Hu, C. (2021, February). Physics-based prognostics of implantable-grade lithium-ion battery for remaining useful life prediction. *Journal of Power Sources*, *485*, 229327. doi: 10.1016/j.jpowsour.2020.229327
- Nectoux, P., Gouriveau, R., Medjaher, K., Ramasso, E., Chebel-Morello, B., Zerhouni, N., & Varnier, C. (2012, June). PRONOSTIA : An experimental platform for bearings accelerated degradation tests. In *IEEE International Conference on Prognostics and Health Management, PHM'12*. (Vol. sur CD ROM, p. 1-8). Denver, Colorado, United States: IEEE Catalog Number : CPF12PHM-CDR.
- Nguyen, A. T., Taniguchi, T., Eciolaza, L., Campos, V., Palhares, R., & Sugeno, M. (2019). Fuzzy control systems: Past, present and future. *IEEE Comput. Intell. Mag*, *14*(1), 56–68. doi: 10.1109/MCI.2018.2881644
- Omariba, Z., Zhang, L., & Sun, D. (2018, May). Review on health management system for lithium-ion batteries of electric vehicles. *Electronics*, *7*(5), 72. doi: 10.3390/electronics7050072
- Qin, P., Zhao, L., & Liu, Z. (2022, March). State of health prediction for lithium-ion battery using a gradient boosting-based data-driven method. *Journal of Energy Storage*, *47*, 103644. doi: 10.1016/j.est.2021.103644
- Richardson, R. R., Birkel, C. R., Osborne, M. A., & Howey, D. A. (2019, January). Gaussian process regression for *In Situ* capacity estimation of lithium-ion batteries. *IEEE Transactions on Industrial Informatics*, *15*(1), 127–138. doi: 10.1109/tii.2018.2794997
- Saha, B., & Goebel, K. (2007). Battery data set. *NASA Ames Prognostics Data Repository*.
- Saha, B., & Goebel, K. (2009). Modeling li-ion battery capacity depletion in a particle filtering framework. In *Proceedings of the annual conference of the prognostics and health management society* (pp. 2909–2924).
- Severiano, C. A., de Lima e Silva, P. C., Cohen, M. W., & Guimarães, F. G. (2021, June). Evolving fuzzy time series for spatio-temporal forecasting in renewable energy systems. *Renewable Energy*, *171*, 764–783. doi: 10.1016/j.renene.2021.02.117
- Shah, J., & Wang, W. (2021, May). An evolving neuro-fuzzy classifier for fault diagnosis of gear systems. *ISA Transactions*. doi: 10.1016/j.isatra.2021.05.019
- Si, X.-S. (2015). An adaptive prognostic approach via nonlinear degradation modeling: Application to battery data. *IEEE Transactions on Industrial Electronics*, *62*(8), 5082–5096. doi: 10.1109/TIE.2015.2393840
- Škrjanc, I. (2021, December). An evolving concept in the identification of an interval fuzzy model of wiener-hammerstein nonlinear dynamic systems. *Information Sciences*, *581*, 73–87. doi: 10.1016/j.ins.2021.09.004
- Soares, E., Angelov, P., & Gu, X. (2020, September). Autonomous learning multiple-model zero-order classifier

- for heart sound classification. *Applied Soft Computing*, 94, 106449. doi: 10.1016/j.asoc.2020.106449
- Wang, L., Pan, C., Liu, L., Cheng, Y., & Zhao, X. (2016, April). On-board state of health estimation of LiFePO₄ battery pack through differential voltage analysis. *Applied Energy*, 168, 465–472. doi: 10.1016/j.apenergy.2016.01.125
- Wu, L., Fu, X., & Guan, Y. (2016, May). Review of the remaining useful life prognostics of vehicle lithium-ion batteries using data-driven methodologies. *Applied Sciences*, 6(6), 166. doi: 10.3390/app6060166
- Xing, Y., Ma, E. W., Tsui, K.-L., & Pecht, M. (2013, June). An ensemble model for predicting the remaining useful performance of lithium-ion batteries. *Microelectronics Reliability*, 53(6), 811–820. doi: 10.1016/j.microrel.2012.12.003
- Zhang, J., Jiang, Y., Li, X., Huo, M., Luo, H., & Yin, S. (2022, June). An adaptive remaining useful life prediction approach for single battery with unlabeled small sample data and parameter uncertainty. *Reliability Engineering & System Safety*, 222, 108357. doi: 10.1016/j.ress.2022.108357
- Zhang, Q., Yang, L., Guo, W., Qiang, J., Peng, C., Li, Q., & Deng, Z. (2022, February). A deep learning method for lithium-ion battery remaining useful life prediction based on sparse segment data via cloud computing system. *Energy*, 241, 122716. doi: 10.1016/j.energy.2021.122716
- Zhang, Y., & Li, Y.-F. (2022, June). Prognostics and health management of lithium-ion battery using deep learning methods: A review. *Renewable and Sustainable Energy Reviews*, 161, 112282. doi: 10.1016/j.rser.2022.112282

Wrong Injection Detection in a Small Diesel Engine, a Machine Learning Approach

Piero Danti¹, Giovanni Vichi², and Ryota Minamino³

^{1,2,3} *Yanmar R&D Europe, Florence, 50125, Italy*

piero.danti@yanmar.com

giovanni.vichi@yanmar.com

ryota.minamino@yanmar.com

¹ *Università degli Studi di Firenze, Florence, 50121, Italy*

piero.danti@yanmar.com

ABSTRACT

In the last ten years, Machine Learning (ML) and Artificial Intelligence (AI) have overwhelmed every engineering research branch finding a broad variety of applications; anomaly detection and anomaly classification are two of the topics that have benefited mostly by data-driven methods' insights. On the other side, in the small diesel engine domain, the current trend is to lean on traditional anomaly detection/classification procedures and do not foster the use of AI. The goal of this work is to detect anomalies in the in-cylinders injectors of a small diesel engine as soon as a wrong quantity of fuel is inputted into one or more cylinders by means of ML approaches. Part of the analysis aim to understand which measurements are the most relevant for the detection and to compare different techniques to select the most suitable one. Furthermore, a condition-based methodology for maintenance is proposed. After a brief review of the state-of-the-art, the case-study scenario is presented grouping sensors accordingly to their degree of accessibility; then, the implemented techniques are explained and results are discussed.

1. INTRODUCTION

Although McCulloch and Pitts proposed the first computational model of a neuron in the 1943 and Arthur Samuel wrote the first computer learning program in 1952, only in the latest years AI has encountered an impressive growth due to the availability of hardware (HW) with an increased computational power and the capability of storing an huge amount of data. In addition, the decrease of sensors and acquisition systems cost and the development of Internet of Things (IoT) infrastructures have led industries to a digital transition.

Piero Danti et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Maintenance, intended as the activity of keeping a machinery or equipment in good condition by making repairs, correcting problems and periodic adjustments, has glimpsed the opportunity of a fast improvement moving from a classical Time-Based Maintenance (TBM) to a more modern Condition-Based Maintenance (CBM).

Many field have demonstrated interest in ML and AI in order to improve their maintenance strategies: Rogers et al. (Rogers, Guo, & Rasmussen, 2019) offer a review of Fault Detection and Diagnosis (FDD) methods for residential air conditioning systems, Datta et al. (Datta & Sarkar, 2016) report different pipeline fault detection methods, Meng et al. (Meng & Li, 2019) investigate Prognostics and Health Management (PHM) methods of lithium-ion batteries, Maciejewski et al. (Maciejewski, Treml, & Flauzino, 2020) deal with fault detection and diagnosis methods for induction motors, Li et al. (Li, Delpha, Diallo, & Migan-Dubois, 2020) study the application of Artificial Neural Networks (ANN) to photovoltaic FDD, Liu et al. (Liu, Yang, Zio, & Chen, 2018) face another blooming subject for AI that is fault detection in rotating machinery while Kumar (Kumar, 2018) takes into account fault detection in a more specific context (bearings and gears), Shi et al. (Shi & O'Brien, 2019) give a comprehensive overview of automated FDD in buildings while Mirnaghi et al. (Mirnaghi & Haghghat, 2020) focus on large-scale HVAC, Gururajapathy et al. review fault location and detection in power distribution systems and, in the end, Habibi et al (Habibi, Howard, & Simani, 2019). are interested in fault detection techniques for wind turbine power generation.

On the other hand, a small number of works has demonstrated interest in detecting and diagnosing faults in Internal Combustion Engines (ICEs) and diesel engines by means of AI and, among these, the majority deals with marine engines: Xu et al. (Xu et al., 2017) propose a new belief rule-based (BRB)

expert system for fault diagnosis, Wang et al. (S. Wang, Wang, & Wang, 2020) assemble a novel scheme for fault diagnosis based on k -means, Principal Component Analysis (PCA) and ANN, Cai et al. design a novel FDD method by combining Rule-Based (RB) algorithm and Bayesian networks (BNs) or Back Propagation Neural Networks (BPNNs), Wei et al. (Wei, Liu, Chen, & Ye, 2020) describe a new unsupervised ML algorithm based on One-Class Support Vector Machine (OCSVM), Affinity Propagation (AP) and Gaussian Mixture Model (GMM) for fault diagnosis, Xu et al. (Xu et al., 2020) define wear fault diagnostic model by using a multi-model fusion system based on Evidential Reasoning (ER), Wang (R. Wang, 2021) use an innovative hybrid fault monitoring scheme integrating the manifold learning and the isolation forest to monitor the state of marine diesel engine.

More specifically, searching works about FDD in small engines (nominal power lower than 155 kW_e, according to YANMAR categorization) has led to unfruitful results; moreover, no paper has shown interest in detecting anomalies concerning the fuel injection into engine's cylinders.

When a diesel engine is calibrated, the quantity of fuel injected in each cylinder is tuned in order to guarantee a certain level of emissions and the desired performance of the system. Due to aging, these values may get uncalibrated and need to be adjusted; usually only a periodical TBM is done and no continuous checks are performed unless the phenomena's evidences are tangible. In this work, a procedure to detect anomalies on the cylinders' injector as soon as a wrong quantity of fuel is inputted in one or more cylinders has been developed; the key questions of this activity are:

- Is it possible to detect a wrong injection anomaly by means of data-driven methods? What kind of accuracy can we expect?
- Is it necessary to install new sensors and which are the most important measurements?

The key findings are:

- Wrong injection can be detected both as general anomaly (at least one cylinder has a wrong injection) and as specific cylinder anomaly with different levels of accuracy.
- Standard sensors acquired by the Engine Control Unit (ECU) are sufficient to train a well performing model but, in case of more stringent requirements, it is highly suggested to measure the cylinder exhaust temperatures.

The paper is structured as follows: after a description of the case-study scenario (Section 2), the ML techniques selected are explained (Section 3), results are discussed (Section 4) and conclusions are drawn (Section 5).

2. CASE-STUDY SCENARIO

Data from a four cylinders YANMAR small diesel engine have been exploited. Each sample presents a certain num-

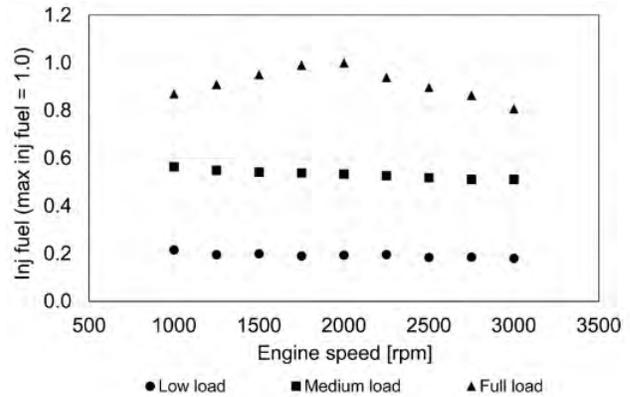


Figure 1. Injected fuel quantity at three load levels by varying the engine speed.

ber of features (inputs to the model) and two labels (outputs of the model) referring respectively to a general anomaly (at least 1 out of 4 cylinders has a wrong injection) and to a localized anomaly (the 1st cylinder presents a wrong injection). Data collection phase was not performed during this activity but was carried out previously; indeed the experimental setup, the sensors description and the engine features are extensively reported by Becciani et al. in (Becciani et al., 2019). For a better understanding Figure 1 shows the matrix of tests performed during data collection; in particular it shows the normalized injected fuel quantity at three engine load levels by varying the engine speed from 1000 rpm to 3000 rpm by steps of 250 rpm.

2.1. Steps definition

Measurements have been split in three groups based on the level of accessibility. In the 1st step, signals directly acquired from the ECU (available from default engine sensors setup) have been considered, then, in the 2nd step, additional sensors installed ad-hoc in the test-bench (but with the possibility to be added at sustainable costs on real engines) have been analysed. The 3rd step involves also sensors for combustion analysis (unlikely to be added on real engines due to the unfeasible costs but interesting from a research point of view).

In particular the sensors groups are defined as follows:

- Step 1 (ECU data):
 - Rail pressure;
 - Intake manifold temperature;
 - Exhaust manifold temperature;
 - Intake manifold pressure;
 - Exhaust manifold pressure.
- Step 2 (test-bench data):
 - Air mass flow;
 - Air-Fuel ratio;
 - Oil pressure and temperature;

- Fuel inlet pressure;
- Exhaust downstream turbine temperature;
- Exhaust downstream turbine pressure;
- Exhaust temperature on cylinders.
- Step 3 (combustion analyser data):
 - Indicated Mean Effective Pressure (IMEP) of cylinders;
 - Max pressure of cylinders;
 - Max pressure angle of cylinders;
 - Burning of 50% of the fuel dose (MBF50) of cylinders.

Actually, in each step additional sensors would have been available but, due to the nature of the data collection phase, some measures had to be neglected.

2.2. Targets definition

As mentioned in the beginning of this section, the available dataset is constituted by two labels:

1. label *Anomaly* refers to a general anomaly, when at least 1 out of the 4 cylinders presents a wrong-injection quantity;
2. label *Cylinder1* refers to a particular anomaly located in the first cylinder. A wrong quantity of fuel has been injected in the first cylinder, no matter what is happening in the others cylinders.

The interest of YANMAR, reported in Table 1, is to detect an *Anomaly* using measurements from step 1 or measurements from Step 1 together with measurements from Step 2. On the other hand, it is interesting to evaluate the detection of *Cylinder1* using measurements from step 1 and 2 or measurements from step 1 and 2 together with measurements from step 3.

Table 1. Targets of the activity with relevant steps.

Sensors from steps	Referred as	Anomaly	Cylinder1
1	Step 1	X	
1 + 2	Step 2	X	X
1 + 2 + 3	Step 3		X

As matter of simplicity, when the authors refer to a model trained with the features belonging to a particular step, it is implicit that also the features from the previous steps have been considered; e.g. a model built using the 2nd step features means that also 1st step features have been used.

2.3. Dataset presentation

The dataset has a matricial shape of $447 \times (n + 2)$, where 447 are the examples populating the dataset, n is the number of features accordingly to the relevant step and 2 are the available labels (as explained in Section 2.2). When dealing with the *Anomaly* label, 1/3 of the examples are normal while

2/3 are anomalous. When considering the *Cylinder1* label, the acquisitions are quite balanced: 55.2% of the examples are anomalous while the 44.8% are normal. These considerations lead to face two classification problems since there are many examples of the anomalous behaviour. The whole dataset has been split keeping the 80%-20% proportion in train-set and test-set; in order to tune hyper-parameters and to perform a fair comparison all algorithms described in Section 3 have been trained and validated using a k -fold cross-validation routine: a procedure that divides a limited dataset into k non-overlapping folds (Hastie, Tibshirani, & Friedman, 2009). In this work k has been set to 3.

3. ALGORITHMS SELECTION

As mentioned in Section 2.3, main properties of the dataset are a small amount of data, a small-medium quantity of features and two boolean target variables: therefore, the most suitable approach is to face a supervised classification problem by means of classical ML techniques. Some of the state-of-the-art statistical techniques to approach a 2-classes classification problem, described in this section, have been selected accordingly to (Hastie et al., 2009) and following authors' experience.

3.1. Linear Discriminant Analysis (LDA)

LDA is a discriminant approach that attempts to model differences among samples assigned to certain groups. The aim of the method is to maximize the ratio of the between-group variance and the within-group variance. When the value of this ratio is at its maximum, then the samples within each group have the smallest possible scatter and the groups are separated from one another the most (Stanimirova, Daszykowski, & Walczak, 2013). The two assumptions that must be fulfilled are the Gaussian distribution and the equal group covariances for the two classes; below (equation 1) the mathematical formulation where x is the vector of features and y is the target variable (0 stays for normal behaviour, 1 for anomaly):

$$\begin{cases} P(x | y = 0) \sim \mathcal{N}(\mu_0, \Sigma_0^2) \\ P(x | y = 1) \sim \mathcal{N}(\mu_1, \Sigma_1^2) \\ \Sigma_0 = \Sigma_1 = \Sigma \end{cases} \quad (1)$$

The algorithm finds a linear decision boundary separating the two classes described by equation 2:

$$2(\Sigma^{-1}(\mu_0 - \mu_1))^T x + (\mu_0 - \mu_1)^T \Sigma^{-1}(\mu_0 - \mu_1) + 2 \ln \left(\frac{P(y = 1 | x)}{P(y = 0 | x)} \right) = 0 \quad (2)$$

3.2. Quadratic Discriminant Analysis (QDA)

QDA is a discriminant approach equivalent to LDA with the exception of the assumption of equal co-variances for the two classes (equation 3) (Ghojogh & Crowley, 2019); these hypotheses lead to a quadratic decision boundary (equation 4).

$$\begin{cases} P(x | y = 0) \sim \mathcal{N}(\mu_0, \Sigma_0^2) \\ P(x | y = 1) \sim \mathcal{N}(\mu_1, \Sigma_1^2) \\ \Sigma_0 \neq \Sigma_1 \end{cases} \quad (3)$$

$$\begin{aligned} x^T(\Sigma_0^{-1} - \Sigma_1^{-1})x + 2(\Sigma_1^{-1}\mu_1 - \Sigma_0^{-1}\mu_0)^T x + \\ + (\mu_1^T \Sigma_1^{-1} \mu_1 - \mu_0^T \Sigma_0^{-1} \mu_0)^T + \\ + 2 \ln \left(\frac{|\Sigma_0|}{|\Sigma_1|} \right) + 2 \ln \left(\frac{P(y = 1 | x)}{P(y = 0 | x)} \right) = 0 \end{aligned} \quad (4)$$

3.3. Ensemble Discriminant Analysis (EDA)

Both LDA and QDA are discriminant algorithms that assign a class predicting the conditional distribution of each class. Analysing equations 2 and 4, it is trivial to note that the component deciding which class to assign is the decision function $\delta(x)$ (equation 5).

$$\delta(x) = \ln \left(\frac{P(y = 1 | x)}{P(y = 0 | x)} \right) \quad (5)$$

$\delta(x)$ can be seen as the confidence of the discriminant algorithm in predicting the class. When $\delta(x) \approx 0$ it means that, according to the algorithm, $P(y = 0 | x) \approx P(y = 1 | x)$ and the confidence is low; when $\delta(x) \gg 0$ the algorithm is confident to predict anomaly, otherwise when $\delta(x) \ll 0$ the algorithm is confident to predict a normal behaviour.

With these assumptions, the authors theorized a new technique coupling LDA and QDA. Below the procedure is explained:

Algorithm 1 EDA algorithm

- train the QDA model on the train-set
 - predict by means of QDA both class \hat{y} and decision function δ
 - check δ for wrong prediction in train-set
 - find δ_{max}^+ and δ_{min}^- to identify a boundary of indecision
 - train an LDA model on the train examples included in the QDA boundary of indecision
 - use the LDA model to predict only the example of test in the QDA boundary of indecision
 - take as final predictions of the QDA boundary of indecision outcomes with the higher decision function value (between the original QDA and the LDA)
-

3.4. k-Nearest Neighbours (kNN)

The intuition underlying kNN Classification is quite straightforward, examples are classified based on the class of their k nearest neighbours. More information can be retrieved in (Cunningham & Delany, 2007).

3.5. Support Vector Machines (SVM)

SVM is a classification technique based on the statistical learning theory proposed by Vapnik (Vapnik, 1999). The objective of the SVM algorithm is to find a hyper-plane in an n -dimensional space, where n is the number of features, that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyper-planes that could be chosen; the SVM algorithm finds a plane that has the maximum margin between the data points of both classes. Maximizing the margin distance provides some robustness to the future classifications. Further information are well explained in (James, Witten, Hastie, & Tibshirani, 2013).

3.6. Classification And Regression Tree (CART)

As the name suggests, CART are suitable both for classification and regression problems. The difference lies in the target variable; this work aims to classify between healthy and anomalous status, indeed the dataset presents a 2-classes target variable. Classification and regression trees are prediction models constructed by recursively partitioning a dataset and fitting a simple model to each partition. Their name derives from the usual practice of describing the partitioning process by a decision tree. For a deeper description and a brief review the reader can refer to (Loh, 2011).

3.7. Artificial Neural Networks (ANN)

Since first years of 21th century, ANN have emerged as an important tool for classification. The vast research activities performed in the last 20 years in neural classification have established that neural networks are a promising alternative to various conventional classification methods. The advantage of neural networks lies in the following theoretical aspects. First, neural networks are data-driven self-adaptive methods in that they can adjust themselves to the data without any explicit specification of functional or distributional form for the underlying model. Second, they are universal functional approximators, indeed ANN can approximate any function with arbitrary accuracy (Zhang, 2000).

3.8. eXtreme Gradient Boosting (XGB)

In (Chen & Guestrin, 2016b), Chen and al. proposes the most used algorithms to solve classification problems in Kaggle competitions. Gradient boosting is a ML technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction

models, typically CART. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function. The extreme in the name refer to the fact that is designed to push the extreme of the computation limits of machines to provide a scalable, portable and accurate library.

4. RESULTS

All algorithms described in Section 3 have been trained (both for predicting *Anomaly* and *Cylinder1*, as summed up in Table 1) on the 80% of the whole dataset using a 3-fold cross-validation routine and then they have been tested on the remaining 20%.

Table 2 reports the algorithms' accuracy for the *Anomaly* case and Table 3 for the *Cylinder1* case; both tables are ordered by the descending accuracy value of the test-set. Accuracy has been used as evaluation metric in view of the fact that the anomalous/healthy samples are balanced within the dataset as explained in Section 2.3.

More complex models, XGB and ANN, show unstable results in the majority of the proposed tests due to the low amount of training examples available, indeed they are prone to over-fitting. On the other hand, CART and kNN appear not feasible to catch engine's non-linearities. Unexpectedly, also SVM seems weak in predicting wrong injections. Focusing on the 2_{nd} step, the most interesting from a business point of view, best performing algorithms are discriminant analysis techniques and in particular QDA and EDA, the new method proposed by the authors.

Considering test-set performance, best algorithms' result for each case can be summarized as follows:

- *Anomaly*, step 1: kNN with a 88% of accuracy;
- *Anomaly*, step 2: EDA and QDA with a 93% of accuracy;
- *Cylinder1*, step 2: EDA and QDA with a 88% of accuracy;
- *Cylinder1*, step 3: ANN with a 91% of accuracy.

Discriminant analysis algorithms demonstrated to perform well in terms of robustness and accuracy; moreover, the decision function values generated by EDA and QDA (equation 5) for the misclassified examples always have a value close to zero. As explained in 3.3, this means that the algorithm claims to be uncertain about its prediction; otherwise, when the examples are well classified, decision function values present high values in module and consequently the algorithm predictions have a high confidence index.

Previous consideration opens the doors to a new perspective, indeed, considering both accuracy and decision function metrics, a new CBM approach can be designed to substitute the classical TBM adopted by engines' manufacturers.

For sake of simplicity, the authors define t the current time,

x_t the vector of features at time t , $\delta(x_t)$ the decision function generated by the discriminant ML algorithm for the features x_t and T_m the injectors maintenance date (used in classical TBM) indicated in the engine's data-sheet; clearly, when a maintenance intervention is performed, the current time t is reset.

The above mentioned CBM approach can be detailed in the following enumeration:

1. for $t < T_m$:
 - when the ML algorithm detects an anomaly with high index of confidence $\delta(x_t)$, a maintenance intervention is needed;
 - when the ML algorithm detects an anomaly with low index of confidence $\delta(x_t)$, the engine can continue to operate normally;
 - when the ML algorithm predicts a normal behaviour no action is needed and the value of $\delta(x_t)$ is not considered.
2. for $t = T_m$ the scheduled TBM intervention is not performed and this casuistry is incorporated in the next point;
3. for $t \geq T_m$:
 - when the ML algorithm detects an anomaly, no matter the value of $\delta(x_t)$, a maintenance intervention is needed;
 - when the ML algorithm foresees a normal behaviour with low index of confidence $\delta(x_t)$, a maintenance intervention is needed;
 - when the ML algorithm foresees a normal behaviour with high index of confidence $\delta(x_t)$, the engine can continue to operate normally.

The engine's manufacturer has the duty to set a coherent threshold for the confidence index $\delta(x_t)$ in order to define the concepts of high and low confidence, minimizing the risks and maximizing customer's profit.

The authors define a general rule to set the threshold on $\delta(x_t)$ as follows:

- in critical applications, where it is highly hazardous to incur in a faulty status, $\delta(x_t) \geq 0.85$ is considered a high index of confidence while $\delta(x_t) < 0.85$ a low index of confidence.
- in non-critical applications, where an anomaly is tolerable, $\delta(x_t) \geq 4.60$ is considered an high index of confidence while $\delta(x_t) < 4.60$ a low index of confidence.

As last consideration, a further analysis has been carried out to understand which measurements are more important to detect a wrong injection. Under a common agreement with the YANMAR engine experts, a feature importance analysis of the step 2 of the *Anomaly* case has been performed since it has been considered the most promising case-study from an application point of view.

Table 2. Accuracy values for algorithms predicting *Anomaly*.

		Anomaly							
<i>step</i>	<i>alg.</i>	kNN	XGB	ANN	LDA	SVM	CART	EDA	QDA
		1							
<i>acc. train</i>		1.000	0.985	0.855	0.750	0.979	0.958	0.763	0.760
<i>acc. val</i>		0.775	0.812	0.781	0.753	0.747	0.736	0.758	0.756
<i>acc. test</i>		0.877	0.843	0.798	0.787	0.787	0.764	0.742	0.742
		2							
<i>step</i>	<i>alg.</i>	EDA	QDA	SVM	LDA	ANN	XGB	kNN	CART
<i>acc. train</i>		0.969	0.938	0.919	0.868	0.987	0.961	1.000	0.878
<i>acc. val</i>		0.947	0.927	0.868	0.868	0.865	0.843	0.784	0.775
<i>acc. test</i>		0.933	0.933	0.876	0.865	0.820	0.798	0.764	0.764

Table 3. Accuracy values for algorithms predicting *Cylinder 1*.

		Cylinder 1							
<i>step</i>	<i>alg.</i>	QDA	EDA	ANN	XGB	SVM	LDA	kNN	CART
		2							
<i>acc. train</i>		0.885	0.896	0.979	1.000	0.952	0.780	1.000	0.801
<i>acc. val</i>		0.868	0.865	0.809	0.820	0.801	0.733	0.767	0.663
<i>acc. test</i>		0.876	0.876	0.854	0.830	0.787	0.775	0.764	0.742
		3							
<i>step</i>	<i>alg.</i>	ANN	QDA	XGB	kNN	SVM	LDA	EDA	CART
<i>acc. train</i>		0.972	0.806	1.000	1.000	0.822	0.787	0.876	0.956
<i>acc. val</i>		0.834	0.748	0.879	0.767	0.767	0.756	0.834	0.739
<i>acc. test</i>		0.910	0.888	0.865	0.831	0.809	0.798	0.778	0.776

Among the best performing algorithms, QDA is the most interpretable. Indeed, analysing the coefficients of the decision boundary (equation 4) quadratic term ($\Sigma_0^{-1} - \Sigma_1^{-1}$), it is straightforward to give a degree of importance to each feature: as physically expected, the cylinders exhaust temperatures are the main drivers to detect a wrong injection (Table 4).

5. DISCUSSION

AI and ML are not extensively used to enhance small diesel engines' performance and, in particular, in-cylinders wrong injection detection or classification usually leans on classical strategies based on physical knowledge; the present work aims to emphasize the exploitability of state-of-the-art ML models to improve commonly used techniques in the ICE domain.

The authors compared various ML algorithms applied to three different groups of features, demonstrating that an high accuracy can be obtained in both the wrong injection classifications investigated: the case of a general *Anomaly* and the case of a particular cylinder anomaly (*Cylinder1*).

Best results have been pursued when detecting an *Anomaly* using as features the measurements collected both from the ECU and from the additional sensors installed in the ad-hoc test bench; in this case the accuracy obtained by the EDA algorithm proposed by the authors reached the value of 95%.

Moreover, also in the less performing tested cases, taking into account the decision function parameter, it is possible to grade an index of confidence that results very low in the misclassified examples. Thus, considering both metrics, accuracy and decision function, the authors proposed a CBM approach to maximize the operative hours of the engine minimizing failures risk.

In the end, a feature importance analysis have been explained detailing the most impacting measurements for detecting an *Anomaly* and, how expected by the domain experts, cylinders exhaust temperatures gained the greatest importance.

As future development, the authors plan to extend the research to other engines in order to validate the discussed results and to apply the *transfer learning* concept. Indeed, the possibility to design a detection methodology based on models trained off-line on data from another engine (with strong similarities) would bring many advantages from a commercial perspective.

Table 4. Analysis of the coefficients of the decision boundary quadratic term ($\Sigma_0^{-1} - \Sigma_1^{-1}$).

	Intake manifold pressure	Fuel inlet pressure	Rail pressure	Exhaust temp. on cyl1	Exhaust temp. on cyl2	Exhaust temp. on cyl3	Exhaust temp. on cyl4	Oil temp.
Intake manifold pressure	-8.0	0.1	6.9	43.5	2.6	-60.9	19.5	1.4
Fuel inlet pressure	0.1	3.3	1.2	23.7	-1.1	-18.9	-1.9	0.5
Rail pressure	6.9	1.2	-9.6	-23.0	-0.1	29.3	-11.8	6.3
Exhaust temp. on cyl1	43.5	23.7	-23.0	409.8	-103.6	-422.7	118.1	12.4
Exhaust temp. on cyl2	2.6	-1.1	-0.1	-103.6	138.1	-6.7	-38.4	9.8
Exhaust temp. on cyl3	-60.9	-18.9	29.3	-422.7	-6.7	695.4	-253.9	-19.9
Exhaust temp. on cyl4	19.5	-1.9	-11.8	118.1	-38.4	-253.9	168.2	1.0
Oil temp.	1.4	0.5	6.3	12.4	9.8	-19.9	1.0	-9.9

NOMENCLATURE

- ANN Artificial Neural Networks
- AP Affinity Propagation
- BN Bayesian Network
- BPNN Back Propagation Neural Network
- BRB Belief Rule-Based
- CART Classification And Regression Tree
- CBM Condition-Based Maintenance
- ECU Engine Control Unit
- EDA Ensemble Discriminant Analysis
- ER Evidential Reasoning
- FDD Fault Detection and Diagnosis
- GMM Gaussian Mixture Model
- HVAC Heating, Ventilation, and Air Conditioning
- ICE Internal Combustion Engine
- IMEP Indicated Mean Effective Pressure
- IoT Internet of Things
- kNN k-Nearest Neighbours
- LDA Linear Discriminant Analysis
- MBF50 Burning of 50% of the fuel dose
- OCSVM One-Class Support Vector Machine
- PCA Principal Component Analysis
- PHM Prognostics and Health Management
- QDA Quadratic Discriminant Analysis
- RB Rule-Based
- SVM Support Vector Machines
- TBM Time-Based Maintenance
- XGB eXtreme Gradient Boosting

REFERENCES

Becciani, M., Romani, L., Vichi, G., Bianchini, A., Asai, G., Minamino, R., ... Ferrara, G. (2019). Innovative control strategies for the diagnosis of injector performance in an internal combustion engine via turbocharger speed. *Energies*, 12.

Chen, T., & Guestrin, C. (2016a). *Xgboost*. <https://xgboost.readthedocs.io/en/stable/>.

Chen, T., & Guestrin, C. (2016b). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM.

Chollet, F., et al. (2015). *Keras*. <https://keras.io>.

Cunningham, P., & Delany, S. (2007). k-nearest neighbour classifiers. *Mult Classif Syst*, 54.

Datta, S., & Sarkar, S. (2016). A review on different pipeline fault detection methods. *Journal of Loss Prevention in the Process Industries*, 41.

Ghojogh, B., & Crowley, M. (2019). Linear and Quadratic Discriminant Analysis: Tutorial. *arXiv:1906.02590 [cs, stat]*.

Habibi, H., Howard, I., & Simani, S. (2019). Reliability improvement of wind turbine power generation using model-based fault detection and fault tolerant control: A review. *Renewable Energy*, 135.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*. Springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: with applications*

in r. Springer.

- Kumar, S. (2018). Condition based maintenance of bearings and gears for fault detection – A review. *Materials Today*.
- Li, B., Delpha, C., Diallo, D., & Migan-Dubois, A. (2020). Application of Artificial Neural Networks to photovoltaic fault detection and diagnosis: A review. *Renewable and Sustainable Energy Reviews*.
- Liu, R., Yang, B., Zio, E., & Chen, X. (2018). Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*, 108.
- Loh, W.-Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1.
- Maciejewski, N. A. R., Trembl, A. E., & Flauzino, R. A. (2020). A Systematic Review of Fault Detection and Diagnosis Methods for Induction Motors. In *2020 FORTEI-International Conference on Electrical Engineering (FORTEI-ICEE)*.
- Meng, H., & Li, Y.-F. (2019). A review on prognostics and health management (PHM) methods of lithium-ion batteries. *Renewable and Sustainable Energy Reviews*, 116, 109405.
- Mirnaghi, M. S., & Haghghat, F. (2020). Fault detection and diagnosis of large-scale HVAC systems in buildings using data-driven methods: A comprehensive review. *Energy and Buildings*, 229.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12.
- Rogers, A., Guo, F., & Rasmussen, B. (2019). A review of fault detection and diagnosis methods for residential air conditioning systems. *Building and Environment*, 161.
- Shi, Z., & O'Brien, W. (2019). Development and implementation of automated fault detection and diagnostics for building systems: A review. *Automation in Construction*, 104.
- Stanimirova, I., Daszykowski, M., & Walczak, B. (2013). Chapter 8 - robust methods in analysis of multivariate food chemistry data. In F. Marini (Ed.), *Chemometrics in food chemistry* (Vol. 28). Elsevier.
- Vapnik, V. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10.
- Wang, R. (2021). Research on the fault monitoring method of marine diesel engines based on the manifold learning and isolation forest. *Applied Ocean Research*.
- Wang, S., Wang, J., & Wang, R. (2020). A novel scheme for intelligent fault diagnosis of marine diesel engine using the multi-information fusion technology. *IOP Conference Series: Materials Science and Engineering*, 782, 032022.
- Wei, Y., Liu, H., Chen, G., & Ye, J. (2020). Fault Diagnosis of Marine Turbocharger System Based on an Unsupervised Algorithm. *Journal of Electrical Engineering & Technology*, 15.
- Xu, X., Yan, X., Sheng, C., Yuan, C., Xu, D., & Yang, J. (2017). A Belief Rule-Based Expert System for Fault Diagnosis of Marine Diesel Engines. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.
- Xu, X., Zhao, Z., Xu, X., Yang, J., Chang, L., Yan, X., & Wang, G. (2020, feb). Machine learning-based wear fault diagnosis for marine diesel engine by fusing multiple data-driven models. *Knowledge-Based Systems*, 190.
- Zhang, P. (2000). Neural networks for classification: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 30.

BIOGRAPHIES

Piero Danti is an Electronic and Automation engineer, he received his Master degree at the University of Florence with a master thesis titled "Coordinated collective motion of a sensor network for optimum target tracking using Extended Kalman Filter" developed at the Technical University of Munich (TUM) in Germany. In June 2011, he joins the french firm Altran where he works as RAMS consultant. In March 2012 he starts to work as Control and Software engineer in the Oil and Gas business for the General Electric supplier Promel. After 3 years of activities, both as PLC/HMI programmer and field service engineer, he has the chance to join the european R&D department of the Japanese company YANMAR where he deepens Machine Learning and Artificial intelligence expertise. His main research fields are ML and AI applied to statistical analysis, anomaly detection and time-series forecasting. Currently he is a PhD candidate at the University of Florence dealing with Machine Learning applied to PHM.

Giovanni Vichi received the degree in Doctor of Philosophy in Energy and Innovative Industrial Technology in April 2013 with a thesis work titled "Numerical-experimental methodologies for the development of highly efficient engines for two wheel vehicles". His main research fields are the one-dimensional engine simulation, engine control, and engine measurement technique. As research fellow, he participated in numerous founded projects. Since 2018 is working for Yanmar R&D Europe, now heading the Engine and Powertrain group.

Minamino Ryota received the Master degree in Social Science of Energy at the Graduate School of Energy Science in Kyoto in 2010. His main research fields are the one-dimensional engine simulation, engine control, and engine measurement technique. In 2010 he joins YANMAR and in 2020 start working for Yanmar R&D Europe as senior researcher in the Engine and Powertrain group.

APPENDIX

6. HYPER-PARAMETERS SELECTION

The analyses presented in this work have been carried out using Python and all algorithms have been imported from the scikit-learn library (Pedregosa et al., 2011) (version 0.24.2) with the exception of ANN that has been assembled using Keras (Chollet et al., 2015) (version 2.7.0) and XGB that is provided by a dedicated library (Chen & Guestrin, 2016a) (version 1.3.3). In particular:

- LDA uses default hyper-parameters;
- QDA uses default hyper-parameters;
- EDA has been designed by authors ensembling LDA and QDA with default parameters;
- kNN hyper-parameters are listed in Table 5;
- SVM hyper-parameters are listed in Table 6;
- CART hyper-parameters are listed in Table 7;
- ANN hyper-parameters are listed in Table 8;
- XGB hyper-parameters are listed in Table 9;

Hyper-parameters not reported in the tables have been set to the relative library’s default value.

Table 5. Best hyper-parameters configuration for kNN algorithm.

step	kNN			
	Anomaly		Cylinder 1	
	1	2	2	3
<i>n_neighbors</i>	1	16	1	1
<i>weights</i>	uniform	distance	distance	distance
<i>algorithm</i>	ball_tree	kd_tree	auto	auto
<i>leaf_size</i>	67	67	30	30
<i>p</i>	4	1	1	1

Table 6. Best hyper-parameters configuration for SVM algorithm

step	SVM			
	Anomaly		Cylinder 1	
	1	2	2	3
<i>C</i>	200	9.73	8.52	500
<i>gamma</i>	5	0.1	0.385	1
<i>kernel</i>	rbf	rbf	rbf	linear

Table 7. Best hyper-parameters configuration for CART algorithm.

step	CART			
	Anomaly		Cylinder 1	
	1	2	2	3
<i>max_depth</i>	None	None	None	None
<i>min_samples_split</i>	2	18	2	2
<i>min_samples_leaf</i>	2	7	6	2
<i>max_leaf_nodes</i>	44	12	12	45
<i>min_impurity_decrease</i>	0	0	0	0
<i>max_features</i>	5	12	3	19
<i>ccp_alpha</i>	0	0	0	0

Table 8. Best hyper-parameters configuration for ANN algorithm.

step	ANN			
	Anomaly		Cylinder 1	
	1	2	2	3
<i>hidden_layers</i>	1	1	1	1
<i>hidden_activation</i>	ReLu	ReLu	ReLu	ReLu
<i>hidden_units</i>	16	16	128	64
<i>output_activation</i>	sigmoid	sigmoid	sigmoid	sigmoid
<i>output_units</i>	1	1	1	1
<i>batch_size</i>	64	64	64	128
<i>learning_rate</i>	0.5	0.5	0.05	0.1
<i>optimizer</i>	Adam	Adam	Adam	Adam

Table 9. Best hyper-parameters configuration for XGB algorithm.

step	XGB			
	Anomaly		Cylinder 1	
	1	2	2	3
<i>n_estimators</i>	50	250	200	150
<i>subsample</i>	0.5	0.5	1	1
<i>colsample_bytree</i>	1	0.5	0.5	0.5
<i>max_depth</i>	7	7	3	7
<i>gamma</i>	0.25	0.5	0.25	0.25
<i>learning_rate</i>	0.25	0.25	0.25	0.25
<i>eval_metric</i>	logloss	logloss	logloss	logloss

Novel Metrics to Evaluate Probabilistic Remaining Useful Life Prognostics with Applications to Turbofan Engines

Ingeborg de Pater, Mihaela Mitici

Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2628HS Delft, The Netherlands

i.i.depater@tudelft.nl

m.a.mitici@tudelft.nl

ABSTRACT

Well-established metrics such as the Root Mean Square Error or the Mean Absolute Error are not suitable to evaluate estimated distributions of the Remaining Useful Life (i.e., probabilistic prognostics). We therefore propose novel metrics to evaluate the quality of probabilistic Remaining Useful Life prognostics. We estimate the distribution of the Remaining Useful Life of turbofan engines using a Convolutional Neural Network with Monte Carlo dropout. The accuracy and sharpness of the obtained probabilistic prognostics are evaluated using the Continuous Ranked Probability Score (CRPS) and weighted CRPS. The reliability of the obtained probabilistic prognostics is evaluated using the α -Coverage and the Reliability Score. The results show that the estimated distributions of the Remaining Useful Life of turbofan engines are accurate, reliable and sharp when using a Convolutional Neural Network with Monte Carlo dropout. In general, the proposed metrics are suitable to evaluate the accuracy, sharpness and reliability of probabilistic Remaining Useful Life prognostics.

1. INTRODUCTION

Maintenance is undergoing a paradigm shift from time-based maintenance, where tasks are scheduled at fixed time intervals, to predictive maintenance. Under predictive maintenance, sensors continuously measure the condition of components. These measurements are used to predict the Remaining Useful Life (RUL) of components. In turn, RUL prognostics are integrated into maintenance planning. Predictive maintenance has the potential to reduce the maintenance costs, while maintaining the reliability of assets (Lee & Mitici, 2020).

Most studies focus on developing *point* RUL prognostics, i.e., one value for the RUL prediction. For example, a prognostic may indicate that the RUL equals 30 flight cycles for an air-

Ingeborg de Pater et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

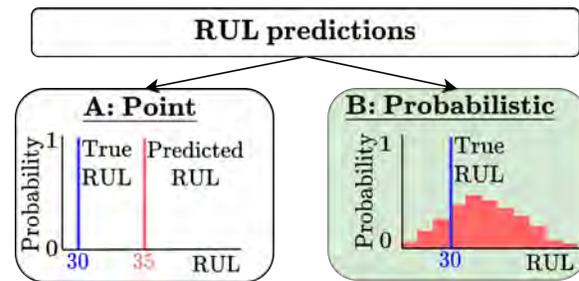


Figure 1. A) Point RUL prognostics, B) Probabilistic RUL prognostics.

craft component (see Figure 1-A). Point RUL prognostics for turbofan engines are developed in (de Pater, Reijns, & Mitici, 2022; Li, Ding, & Sun, 2018) using a Convolutional Neural Network (CNN) and in (Xia, Feng, Lu, Fei, & Xue, 2021) using a Long Short-Term Memory neural network. In (Mitici & de Pater, 2021), point RUL prognostics for aircraft Cooling Units are developed using particle filtering. In (Lee & Mitici, 2022), point RUL prognostics are obtained for aircraft landing gear brakes using linear regression.

For reliability purposes, however, it is key that the uncertainty associated with the estimated RUL is also determined. In this line, several studies estimate the distribution of RUL, i.e., probabilistic RUL prognostics (see Figure 1-B). In (Nguyen & Medjaher, 2019) and (Biggio, Wieland, Chao, Kastanis, & Fink, 2021) the RUL distribution of turbofan engines is obtained using a Long Short-Term Memory neural network and Deep Gaussian processes, respectively. In (de Pater & Mitici, 2021) the RUL distribution of aircraft Cooling Units is estimated using particle filtering. Probabilistic RUL prognostics for nuclear components are developed in (Baraldi, Mangili, & Zio, 2015) using Gaussian Process regression. Last, in (Le Son, Fouladirad, & Barros, 2016) the RUL distribution is estimated using a noisy Gamma deterioration process.

To evaluate probabilistic RUL prognostics, well-established metrics such as the Root Mean Square Error (RMSE) or the Absolute Mean Error (MAE) are not directly applicable. In

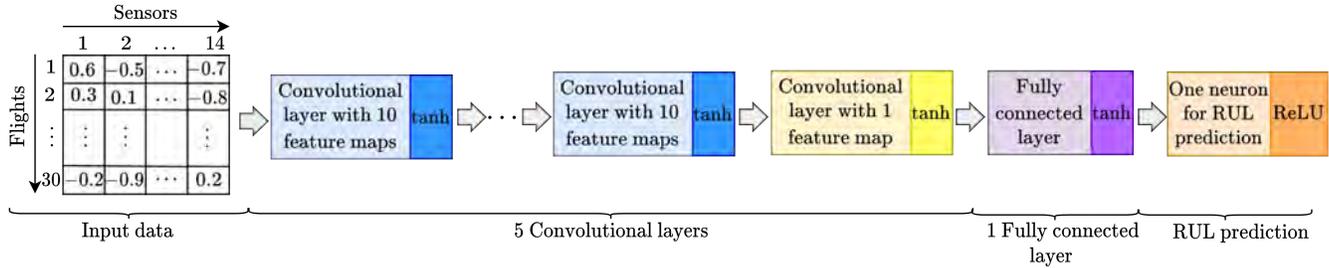


Figure 2. Schematic overview of the CNN architecture for dataset FD001.

principle, RMSE and MAE could be computed relative to the mean of the estimated RUL distribution. However, this would disregard the variance and sharpness of the estimates, and give little indication of the actual trustworthiness of the RUL prognostics. In (Saxena et al., 2008; Saxena, Celaya, Saha, Saha, & Goebel, 2009), a few metrics are proposed to evaluate probabilistic RUL prognostics such as prognostic horizon, probabilistic $\alpha - \lambda$, (cumulative) relative accuracy and convergence. These metrics evaluate the accuracy of the RUL prognostics, and specifically on how this accuracy changes over the lifetime of components. For an example of their usage, see (Lall, Lowe, & Goebel, 2011). However, these metrics all require a sequence of RUL prognostics over the lifetime of each component. Yet, for many publicly available degradation test sets, such as the C-MAPSS data set on turbofan engines (Saxena & Goebel, 2008), only one RUL prognostic per test instance can be determined. As such, the prognostic horizon, probabilistic $\alpha - \lambda$, relative accuracy and convergence cannot be used to evaluate these single probabilistic RUL prognostics. Most importantly, these metrics do not explicitly quantify the reliability of the probabilistic RUL prognostics.

In this paper we propose novel metrics to evaluate the accuracy and sharpness of probabilistic RUL prognostics (CRPS and weighted CRPS), and metrics to explicitly evaluate the reliability (α -Coverage and Reliability Score). Compared with existing metrics, the weighted CRPS uses penalties when the RUL is overestimated/underestimated. Depending on the type of component, these penalties can be adjusted. For example, for safety critical components it is important that the RUL is not overestimated. Otherwise, an overestimated RUL could lead to a missed failure. In such cases, the weighted CRPS applies a larger penalty for a RUL overestimation than for a RUL underestimation. The Reliability Diagram and Reliability Score provide a means to graphically visualize the performance of the RUL prognostics. Unlike existing numerical metrics, this metric provides a visual interpretation of the performance of the prognostics as well. We illustrate our metrics for probabilistic RUL prognostics for the turbofan engines of the C-MAPSS data set. Here, we estimate a distribution of the RUL of the turbofan engines using a Convolutional Neu-

ral Network with Monte Carlo dropout.

In Section 2, we introduce the Convolutional Neural Network with Monte Carlo dropout to estimate a RUL probability distribution for turbofan engines. We next propose metrics to evaluate these estimated RUL distributions in Section 3. We illustrate the proposed metrics in a case study in Section 4.

2. PROBABILISTIC RUL PROGNOSTICS FOR TURBOFAN ENGINES USING A CONVOLUTIONAL NEURAL NETWORK WITH MONTE CARLO DROPOUT

In this section, we generate *probabilistic* RUL prognostics for aircraft turbofan engines using a Convolutional Neural Network (CNN) and Monte Carlo dropout. Specifically, we estimate the probability density function (pdf) of the RUL of an engine, and not just one point value for the RUL. We apply our methodology to the turbofan engine degradation simulation C-MAPSS dataset, which is generated using the NASA Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) program (Saxena & Goebel, 2008). The dataset contains measurements of 21 sensors that monitor the degradation of the turbofan engines. The C-MAPSS dataset consists of four data subsets, each with a different number of operational and fault conditions (see Table 1). Each subset contains a training set, with run-to-failure instances, and a test set. For each failure instance in the test set, the data is terminated somewhere before failure with the aim to predict the RUL. More information on this publicly available data set can be found in (Ramasso & Saxena, 2014).

Table 1. C-MAPSS datasets for turbofan engines.

	FD001	FD002	FD003	FD004
Training instances	100	260	100	249
Testing instances	100	259	100	248
Operating conditions	1	6	1	6
Fault conditions	1	1	2	2

We select 14 out of the 21 sensors available from C-MAPSS that have non-constant measurements. The remaining 7 sensors exhibit constant measurements and are thus not considered for RUL prediction. The selected sensor measurements

are normalized using min-max normalization (Li et al., 2018) with respect to the operating condition (Babu, Zhao, & Li, 2016). We also include the history of the operating conditions in the input of the CNN, i.e., the number of flights spent in each operating condition, as in (Babu et al., 2016).

The architecture and hyperparameters of the CNN are similar to the CNN proposed in (Li et al., 2018). Specifically, the CNN consists of 5 convolutional layers, where the first four convolutional layers each have 10 kernels of size 10×1 (i.e., one-dimensional kernels). The last convolutional layer has one kernel of size 3×1 , combining all 10 feature maps into one feature map. This last feature map is flattened in a flatten layer, and connected to a fully connected layer. All these layers use the tangent (tanh) activation function. Last, one single neuron is attached to the fully connected layer to predict the RUL using the Rectified Linear Unit (ReLU) activation function. A schematic overview of this CNN is in Figure 2. The weights of the CNN are optimized using the Adam optimizer (Kingma & Ba, 2014) with a batch size of 512 samples, and a maximum of 250 training epochs. The learning rate is 0.001 for the first 200 epochs, and 0.0001 for the last 50 epochs. A cut-off value R_{early} of 125 flights is applied. We use a window size of 30 flights for FD001 and FD003, of 20 flights for FD002 and of 15 flights for FD004.

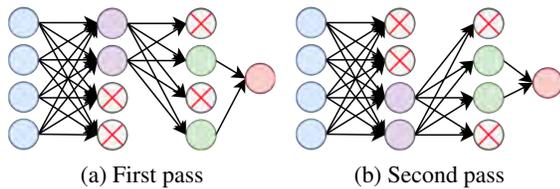


Figure 3. Monte Carlo dropout during two different passes through the network, in a neural network with two fully connected layers.

To obtain a probability distribution of the RUL using CNN, we additionally apply Monte Carlo dropout (Biggio et al., 2021; Gal & Ghahramani, 2016). During the training phase, we apply a dropout rate of $\rho = 0.5$ in each layer, with the exception of the last convolutional layer before the flatten layer, and the first convolutional layer (Gal, Hron, & Kendall, 2017). During the testing phase, we also use dropout and predict the RUL of each test instance i for $M_i > 1$ times, each time randomly selecting neurons to be dropped. This is illustrated in Figure 3. The pdf of the RUL for a test instance i is now created with the M_i RUL predictions.

Figure 4 shows the obtained pdf of the RUL for engines $i \in \{53, 4, 86, 67\}$ of test set FD001. The pdf of the RUL of engine 53 is well centered around the actual RUL, and the variance is relatively low. The pdf of the RUL of engine 4 is well centered around the actual RUL as well, but the variance is larger, suggesting a larger uncertainty about the prediction. In contrast, the pdf's of the RUL of engines 86 and 67 are not

well centered around the actual RUL. Moreover, the actual RUL of engine 67 falls outside the estimated RUL probability distribution.

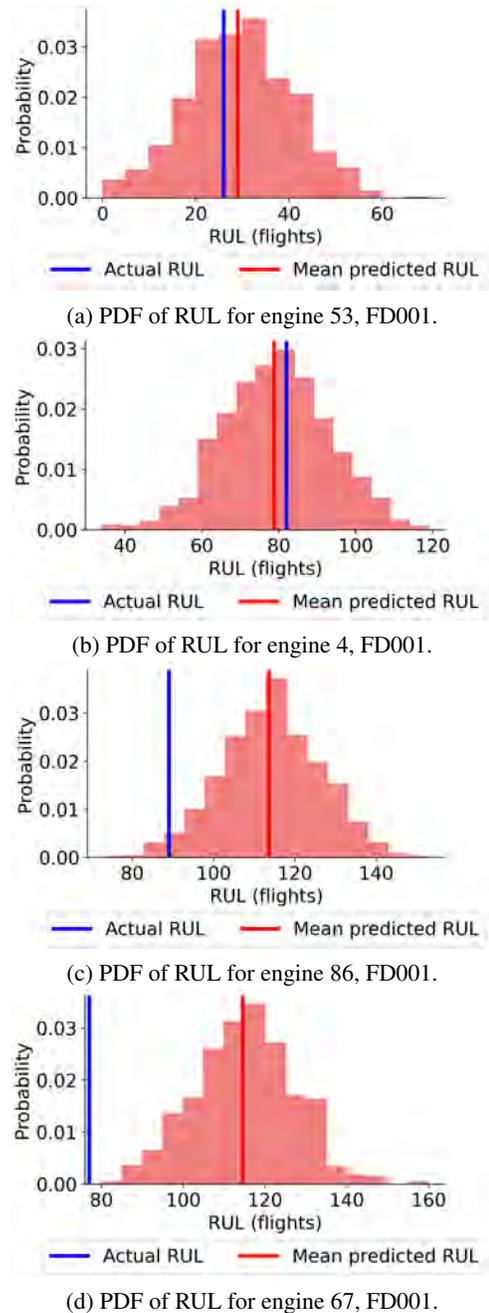


Figure 4. The estimated pdf of the RUL of four individual engines in the test set of FD001.

2.1. Metrics often used to evaluate RUL prognostics

The metrics often used to assess the performance of point RUL prognostics are the Mean Absolute Error (MAE), the Root Mean Square Error (RMSE) and the Mean Score. These

metrics are computed based on the actual RUL vs. the predicted point RUL. When the pdf of the RUL is estimated instead, the MAE, RMSE and the Mean Score can be computed based on the actual RUL vs. the *mean* of the predicted RUL.

Formally, let N be the number of test instances in one C-MAPSS test set, and let y_i be the actual RUL for test instance i . Let \hat{y}_{ij} , $j \in \{1, 2, \dots, M_i\}$, be the j^{th} RUL prediction for engine i . Let \bar{y}_i be the mean predicted RUL of test instance i :

$$\bar{y}_i = \frac{1}{M_i} \sum_{j=1}^{M_i} \hat{y}_{ij}. \quad (1)$$

Then, when considering probabilistic RUL prognostics,

$$\text{MAE}^p = \frac{1}{N} \sum_{i=1}^N |\bar{y}_i - y_i|. \quad (2)$$

$$\text{RMSE}^p = \sqrt{\frac{1}{N} \sum_{i=1}^N (\bar{y}_i - y_i)^2}. \quad (3)$$

$$\text{Mean Score}^p = \frac{1}{N} \sum_{i=1}^N s_i, \quad (4)$$

with

$$s_i = \begin{cases} e^{-\frac{\bar{y}_i - y_i}{\gamma}} - 1, & \bar{y}_i - y_i < 0 \\ e^{\frac{\bar{y}_i - y_i}{\delta}} - 1, & \bar{y}_i - y_i \geq 0 \end{cases},$$

with γ and δ user-defined metrics. For the C-MAPSS data set, $\gamma = 13$ and $\delta = 10$ are usually applied (Li et al., 2018).

Table 2. RMSE^p, MAE^p, and Mean Score^p with respect to the mean RUL prediction - C-MAPSS dataset.

Test set	RMSE ^p	MAE ^p	Mean Score ^p
FD001	12.76	9.22	2.78
FD002	14.74	11.14	3.55
FD003	11.89	9.07	2.43
FD004	18.03	13.44	8.03

Table 2 shows the RMSE^p, MAE^p and Mean Score^p obtained for our probabilistic RUL prognostics when using the C-MAPSS dataset and a CNN with Monte Carlo dropout. Training the neural network took between 12.1 (FD001) to 27.3 (FD002) seconds per epoch on a computer with an Intel Core i7 processor at 2.11 GHz and 8Gb RAM. Our results are comparable with state-of-the-art RUL prognostic results in (Xia et al., 2021).

However, these metrics do not fully capture the quality of the probabilistic RUL prognostics. The reliability and sharpness of the RUL prognostics is not evaluated, e.g, the variance of the generated pdf of the RUL. For example, for engine 4 (see Figure 4b) the absolute error with the mean predicted RUL is only 3.2 flights, and the Score with the mean predicted RUL

is only 0.28. The mean predicted RUL is thus very close to the actual RUL. However, the standard deviation of the pdf of the RUL is large ($\sigma = 13.6$), suggesting a large uncertainty in the prediction. This large variance is not reflected in the mean predicted RUL, and thus neither in the RMSE^p, MAE^p and Mean Score^p metrics. Similarly, for engine 86 (see Figure 4c), the absolute error with the mean predicted RUL is 24.6 flights, and the score value with the mean predicted RUL is 10.67, which shows that the mean predicted RUL is far off the actual RUL. However, the actual RUL still falls within the pdf of the RUL. This is again not reflected in the mean RUL prediction and thus in the three metrics above. To analyze the full predicted pdf of the RUL with the corresponding uncertainty estimates, we introduce four additional metrics that characterize the reliability, the sharpness and the accuracy associated with the pdf's of the RUL.

3. NOVEL METRICS TO EVALUATE PROBABILISTIC RUL PROGNOSTICS

In this section, we introduce the following novel metrics to characterize the reliability, the sharpness and the accuracy of probabilistic RUL prognostics (i.e, when estimating the pdf of the RUL): the Continuous Ranked Probability Score (CRPS), the weighted CRPS (CRPS^W), the α -Coverage, and the Reliability Score (RS). In the appendix, we provide the Python code to calculate the proposed metrics.

3.1. Continuous Ranked Probability Score (CRPS)

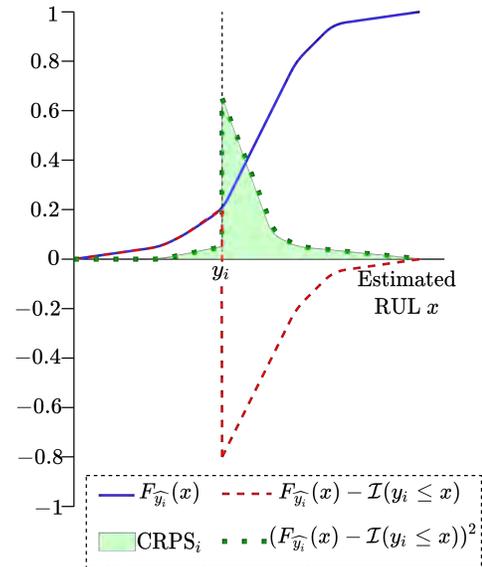


Figure 5. Illustration of the CRPS_{*i*} metric for a single component i .

The Continuous Ranked Probability Score (CRPS) evaluates i) if the estimated RUL distribution is centered around the actual RUL of a component i , i.e., the accuracy of the RUL

prognostic, and ii) if the variance of the RUL distribution is low, i.e., the sharpness of the RUL prognostic. In other words, a probabilistic RUL prognostic for a component i is best when all RUL predictions $\hat{y}_{ij} \ j \in \{1, 2, \dots, M_i\}$ are close to the actual RUL y_i .

CRPS has been used to evaluate probabilistic predictions for applications such as flight delays (Zoutendijk & Mitici, 2021), sea level pressure and surface temperature (Gneiting, Raftery, Westveld III, & Goldman, 2005) and electricity prices (Nowotarski & Weron, 2018). However, to the best of our knowledge, this metric has not yet been used to evaluate probabilistic RUL prognostics.

Let $F_{\hat{y}_i}(x)$ denote the estimated, empirical CDF of the RUL of a component i . Then CRPS is as follows:

$$\begin{aligned} \text{CRPS} &= \frac{1}{N} \sum_{i=1}^N \text{CRPS}_i, \quad (5) \\ \text{CRPS}_i &= \int_{-\infty}^{\infty} (F_{\hat{y}_i}(x) - \mathcal{I}\{y_i \leq x\})^2 dx, \\ \text{with } \mathcal{I}\{y_i \leq x\} &= \begin{cases} 1, & y_i \leq x \\ 0, & y_i > x. \end{cases} \end{aligned}$$

Intuitively, CRPS for a component i can be seen as a probabilistic generalization of the absolute error $|y_i - \hat{y}_i|$. Specifically, when calculating the CRPS of a point RUL prediction, we obtain the absolute error of this point RUL prediction. The smaller the CRPS metric is, the closer the RUL prediction is to the actual RUL. In an ideal case when a perfect RUL prediction without uncertainty (i.e., a point RUL prediction) is obtained, CRPS equals zero. A comprehensive explanation of this metric can be found in (Gneiting & Katzfuss, 2014).

Figure 5 shows a graphical representation of CRPS for a single, generic component i . The blue, solid line represents the empirical CDF of the RUL prognostic of this component i . The light-green area is the CRPS for this component i . This area (i.e., the CRPS value) is small if the accuracy and sharpness of the probabilistic RUL prognostic are high. In general, if the prognostics are accurate, then most RUL estimates are located close to the true RUL y_i . This is equivalent to a low CRPS value. If the prognostics are not only accurate, but also sharp, then the tails of the distribution are small and low. In this case, the CRPS value is smaller as well. Conversely, the CRPS value increases if the true RUL y_i falls outside the estimated RUL distribution (i.e., inaccurate prognostics).

3.2. Weighted CRPS (CRPS^W)

For most components and systems, overestimating the RUL is much more detrimental than underestimating the RUL (Li et al., 2018). A late prediction of the failure time is less desirable since missing a component failure may have severer consequences than replacing this component too early. We

thus propose the weighted CRPS, which considers penalties for the RUL being overestimated/underestimated. Depending on the type of component, these penalties can be adjusted. In the case of safety critical component, for example, larger penalties are applied for RUL being overestimated. This is because a RUL overestimation may lead to a missed failure. The weighted CRPS is defined as follows:

$$\begin{aligned} \text{CRPS}^W &= \frac{1}{N} \sum_{i=1}^N \text{CRPS}_i^W, \quad (6) \\ \text{CRPS}_i^W &= (2 - \beta) \int_{-\infty}^{y_i} (F_{\hat{y}_i}(x) - \mathcal{I}\{y_i \leq x\})^2 dx \\ &\quad + \beta \int_{y_i}^{\infty} (F_{\hat{y}_i}(x) - \mathcal{I}\{y_i \leq x\})^2 dx, \\ &= (2 - \beta) \int_{-\infty}^{y_i} (F_{\hat{y}_i}(x))^2 dx + \beta \int_{y_i}^{\infty} (F_{\hat{y}_i}(x) - 1)^2 dx, \end{aligned}$$

with $0 \leq \beta \leq 2$ an user-specific parameter. The magnitude of the penalty is specified through the weight β . The weight β is specified by the user, and depends on the domain application.

3.3. α -Coverage

CRPS evaluates the accuracy and sharpness of the probabilistic RUL prognostics. It is, however, also important to verify the reliability of the RUL predictions. To address this, we introduce the coverage of a RUL prediction, similar to (Baraldi et al., 2015). In this paper, however, we construct the coverage of a probabilistic RUL prognostic without assuming that this prognostic follows a specific distribution, such as the Gaussian distribution.

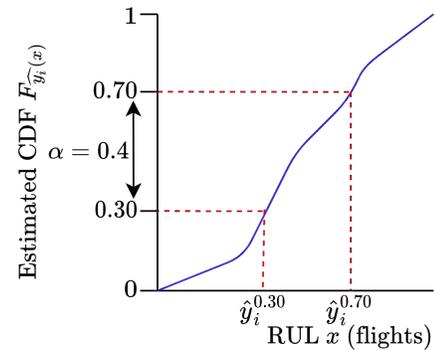


Figure 6. Illustration of the percentiles with the estimated CDF of the RUL of a test instance i

To calculate the coverage, we first construct a credible interval around the median of the estimated RUL distribution with width α . For example, let us assume that we have $M_i = 1000$ RUL predictions for a test instance i , i.e., $\hat{y}_{ij}, j \in \{1, 2, \dots, M_i\}$. Let us consider the credible interval around the median

with width $\alpha = 0.4 = 40\%$. Then, this credible interval is $[\hat{y}_i^{0.30}, \hat{y}_i^{0.70}]$, with $\hat{y}_i^{0.30}$ the RUL prediction belonging to the $50\% - 0.5\alpha = 30^{\text{th}}$ percentile. In our example, when we sort all $M_i = 1000$ predictions from small to large, this is the $j = 300^{\text{th}}$ RUL prediction $\hat{y}_{i,300}$. Also, $\hat{y}_i^{0.70}$ is the RUL prediction belonging to the $50\% + 0.5\alpha = 0.70^{\text{th}}$ percentile. In our example, when we sort all $M_i = 1000$ predictions from small to large, this is the $j = 700^{\text{th}}$ RUL prediction $\hat{y}_{i,700}$. The predicted probability that the actual RUL y_i of component i is within the credible interval $[\hat{y}_i^{0.30}, \hat{y}_i^{0.70}]$ is $\alpha = 40\%$. This example is illustrated in Figure 6.

We construct a credible interval with width $\alpha = 0.4$ for all $i \in \{1, 2, \dots, N\}$ test instances. It is expected that for $\alpha = 40\%$ of the test instances, the actual RUL y_i is within the credible interval $[\hat{y}_i^{0.30}, \hat{y}_i^{0.70}]$. If the actual RUL of more than 40% of the test instances falls within the credible interval $[\hat{y}_i^{0.30}, \hat{y}_i^{0.70}]$, then the *uncertainty* for $\alpha = 0.4$ is *overestimated*. Otherwise, the *uncertainty* for $\alpha = 0.4$ is *underestimated*.

With the concept of a credible interval, the coverage of probabilistic RUL prognostics is defined as follows:

$$\alpha\text{-Coverage} = \frac{1}{N} \sum_{i=1}^N \mathcal{I}(\alpha)_i, \quad (7)$$

$$\text{with } \mathcal{I}(\alpha)_i = \begin{cases} 1, & y_i \in [\hat{y}_i^{0.5-0.5\alpha}, \hat{y}_i^{0.5+0.5\alpha}] \\ 0, & \text{Otherwise,} \end{cases}$$

where $\alpha \in [0, 1]$ is a user-defined parameter and \hat{y}_i^k is the RUL prediction of the k^{th} percentile of the estimated RUL distribution of component i . The closer the coverage is to α , the more reliable the estimated RUL distribution is. The uncertainty is overestimated if the coverage is larger than α . Conversely, the uncertainty is underestimated if the coverage is smaller than α . For example, in Figure 4c, the true RUL does not fall within the 90% credible interval of the RUL distribution. If we predict a RUL distribution for ten individual components, we expect that for only one out of these ten components, the true RUL lies outside the 90% credible interval, as is the case in Figure 4c.

Last, if two RUL prediction methods have the same coverage for a width α , the method that provides tighter credible intervals is preferred. In other words, a higher sharpness of the RUL distributions is preferred. In this way, the predicted RUL distributions give a more precise picture of the actual RUL. A higher sharpness also leads to a lower CRPS. The tightness of the credible intervals, or the mean width of the credible intervals, is defined as (Baraldi et al., 2015):

$$\alpha\text{-Mean width} = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i^{0.5+0.5\alpha} - \hat{y}_i^{0.5-0.5\alpha}). \quad (8)$$

3.4. Reliability Score (RS)

Though the Coverage metric indicates the reliability of the estimated RUL distribution, this reliability is evaluated only relative to a specific α . To conduct a generic, parameter-free reliability analysis of the estimated RUL distribution, we next introduce the Reliability Score (RS). We first introduce the concept of the reliability diagram.

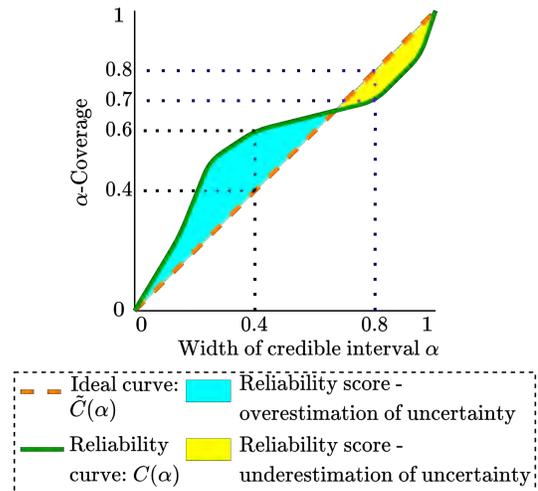


Figure 7. Illustration of the reliability diagram and the Reliability Scores.

For classification problems, a reliability diagram is used as a visual representation of the reliability of the uncertainty associated with the predictions. A reliability diagram is also referred to as a calibration curve. In (Saxena et al., 2008), the reliability diagram is proposed as a RUL prognostic metric. Here, the problem of RUL prognostics is posed as a classification problem with multiple classes. In contrast, in (Vandal, Livingston, Piho, & Zimmerman, 2018), the reliability diagram is defined based on the concept of coverage (see Section 3.3). In doing so, a regression problem does not have to be posed as a multi-class classification problem to construct a reliability diagram. The authors of (Vandal et al., 2018) determine a reliability diagram for flight delay estimations. Similarly, we define a reliability curve $C(\alpha)$ based on α -Coverage (see Eq. (7)) for probabilistic RUL prognostics, i.e., $C(\alpha) = \{\alpha\text{-Coverage}, \alpha \in \{0.00, 0.01, 0.02, \dots, 1.00\}\}$. The reliability diagram is then a visual representation of this reliability curve. Figure 7 gives an illustration of a reliability curve.

The reliability diagram is used to visually inspect whether the uncertainty associated with the RUL predictions is over- or underestimated. For example, when $\alpha = 0.4$, the ideal coverage would be 0.4 as well. In this case, the actual RUL of 40% of the test instances would fall inside a credible interval with width $\alpha = 0.4$. However, in the example in Figure 7, the 0.4-Coverage is 0.6, i.e., the actual RUL of 60% of the

test instances falls inside the credible interval, instead of 40% of the test instances. The uncertainty of the RUL prognostics is thus overestimated.

In contrast, the uncertainty of the RUL prognostics is underestimated at $\alpha = 0.8$ in Figure 7. Here, the actual RUL of only 70% of the test instances falls inside the credible interval with a width of $\alpha = 0.8$.

In general, for classification problems, the Brier Score (Brier, 1950) is used to quantify the reliability of predictions. However, in our adaption of the reliability diagram, each test instance may fall into multiple credible intervals. The calculation of the Brier Score is thus not directly applicable. To address this, we define the following Reliability scores (RS) to quantify the reliability of the RUL prognostics:

$$RS^{\text{under}} = \int_0^1 \mathcal{I}\{C(\alpha) \leq \alpha\}(\alpha - C(\alpha))d\alpha, \quad (9)$$

$$RS^{\text{over}} = \int_0^1 (1 - \mathcal{I}\{C(\alpha) \leq \alpha\})(C(\alpha) - \alpha)d\alpha, \quad (10)$$

$$RS^{\text{total}} = RS^{\text{under}} + RS^{\text{over}}, \quad (11)$$

$$\text{with } \mathcal{I}\{C(\alpha) \leq \alpha\} = \begin{cases} 1, & C(\alpha) \leq \alpha \\ 0, & \text{Otherwise} \end{cases}.$$

The RS^{over} quantifies the overestimation and RS^{under} the underestimation of the *uncertainty* associated with the probabilistic RUL prognostics. Let $\tilde{C}(\alpha) = \{\alpha, \alpha \in \{0.00, 0.01, \dots, 1.00\}\}$ be the ideal curve, i.e., the curve where the Coverage is exactly the width of the credible interval α . To quantify the extent to which the uncertainty associated with the probabilistic RUL prognostics is *underestimated*, we calculate the area RS^{under} between the ideal curve and the reliability curve $C(\alpha)$ when the reliability curve is *below* the ideal curve (i.e., $C(\alpha) \leq \alpha$, yellow area in Figure 7). To quantify the extent to which the uncertainty associated with the probabilistic RUL prognostics is *overestimated*, we calculate the area RS^{over} between the ideal curve and the reliability curve $C(\alpha)$ when the reliability diagram is *above* the ideal curve (i.e., $C(\alpha) \geq \alpha$, blue area in Figure 7). The total RS (RS^{total}) is then the sum of RS^{over} and RS^{under} .

4. RESULTS

In this section, we evaluate our metrics for the obtained probabilistic RUL prognostics for the turbofan engines in the C-MAPSS dataset. These probabilistic RUL prognostics are obtained with a CNN with Monte Carlo Dropout (see Section 2). Figure 9 shows the obtained probabilistic RUL prognostics, and Table 3 shows the corresponding values of the four proposed metrics. The CRPS is lowest for data subset FD003 (6.56), and highest for data subset FD004 (10.09). This is in line with the obtained MAE, which is also lowest for data subset FD003 and highest for data subset FD004. CRPS thus

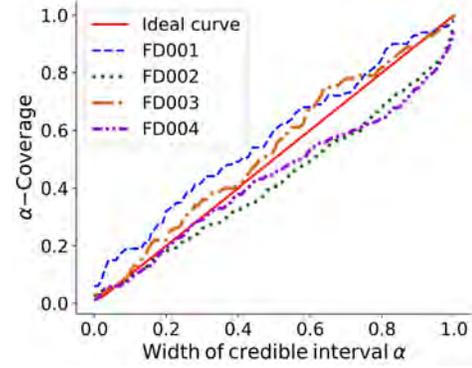


Figure 8. Reliability diagrams - C-MAPSS data subsets.

gives a good overview of the general performance of probabilistic RUL prognostics. Moreover, in contrast with MAE, the sharpness and accuracy of the estimated RUL distributions are also reflected by the CRPS values.

For data subset FD002, the $CRPS^W$ is lower than the CRPS. This indicates that for this dataset, the RUL is relatively often underestimated. In contrast, for data subset FD003, the $CRPS^W$ is higher than the CRPS. This indicates that for FD003, the RUL is relatively often overestimated. The weighted CRPS, compared to the standard CRPS, thus gives a good indication on whether the RUL is usually over- or underestimated.

The reliability diagram of the four data subsets is shown in Figure 8. For data subsets FD001 and FD003, the uncertainty of the RUL prognostics is slightly overestimated. In other words, the prognostics indicate that the RUL lies in an interval with a certain probability. However, these probabilities are too small, relative to the actual number of times the RUL falls within these intervals. For example, let us consider the 0.5-Coverage of data subset FD001. Here, the estimated probability that a test instance falls inside its credible interval with width 0.5 equals 0.5. We thus expect that 50% of the test instances fall inside their credible interval with width 0.5, and 50% fall outside their credible interval. However, 60% of the test instances fall inside their credible interval with width 0.5, i.e., the observed probability is 0.6 instead of 0.5. This shows that the uncertainty associated with the RUL estimates is overestimated. In contrast, for data subsets FD002 and FD004, the uncertainty is underestimated, i.e., the prognostics indicate that the RUL lies in an interval with a certain probability. These probabilities are too high relative to the actual number of times the RUL falls within these intervals. Table 3 shows that the over- and underestimation of the uncertainty associated with the RUL prognostics is well quantified by the reliability scores.

Table 3 also shows the 0.5-Coverage and the 0.95-Coverage. Also this metric indicates that the RUL prognostics for data subsets FD001 and FD003 overestimate the uncertainty associated with the prognostics, while for data subsets FD002

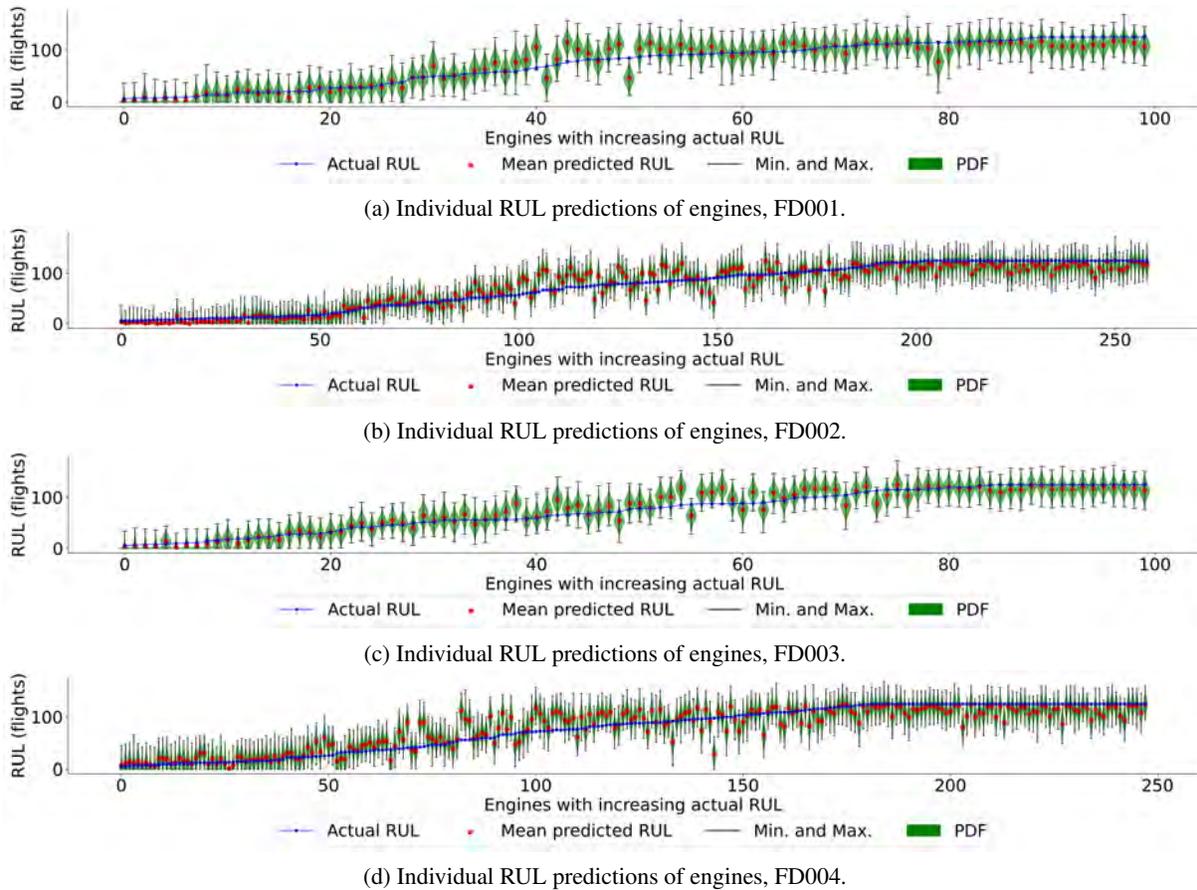


Figure 9. Mean predicted RUL and associated RUL distribution - FD001, FD002, FD003, FD004 of the C-MAPSS test sets.

Table 3. Results for the four C-MAPSS data sets with respect to the uncertainty estimation.

Test set	CRPS ^W		Coverage- $\alpha = 0.5$	Mean width- $\alpha = 0.5$		Coverage- $\alpha = 0.95$	Mean width- $\alpha = 0.95$		RS ^{over}	RS ^{under}	RS ^{total}
	MAE ^p	CRPS		$(\beta = 1.5)$							
FD001	9.22	6.97	7.03	0.60	16.9	0.95	48.0	0.073	0.001	0.074	
FD002	11.14	8.44	7.80	0.40	13.2	0.83	38.0	0.001	0.077	0.078	
FD003	9.07	6.56	7.27	0.53	15.4	0.93	44.7	0.034	0.001	0.035	
FD004	13.44	10.09	10.38	0.44	15.5	0.81	43.8	0.001	0.065	0.065	

Table 4. Performance metrics for engines 53, 4, 86 and 67 in the test set of FD001.

Engine number i	Actual RUL y_i (flights)	Mean predicted RUL \bar{y}_i (flights)	Error: $y_i - \bar{y}_i$ (flights)	Score ^p s_i	CRPS _{i} ^W		$\mathcal{I}(\alpha)_i$ $\alpha = 0.5$	$\hat{y}_i^{0.75} - \hat{y}_i^{0.25}$	$\mathcal{I}(\alpha)_i$ $\alpha = 0.95$	$\hat{y}_i^{0.975} - \hat{y}_i^{0.025}$
					$\beta = 1.5$					
53	26	29.0	-3.0	0.35	2.96	3.72	1	15	1	45
4	82	78.8	3.2	0.28	3.49	2.68	1	19	1	54
86	89	113.6	-24.6	10.67	17.98	26.96	0	16	1	48
67	77	114.5	-37.5	41.61	30.74	46.11	0	16	0	46

and FD004 the uncertainty associated with the prognostics is underestimated. Moreover, the mean width of the 0.95 credible interval is large, ranging from 38.0 flights (data subset FD002) to 48.0 flights (data subset FD001), i.e., the sharpness of the RUL distributions is low.

4.1. RUL prognostics for individual engines

In this section, we analyze our proposed metrics for probabilistic RUL prognostics for four specific engines 53, 4, 86 and 67 of data subset FD001, see Table 4. The probabilistic RUL prognostics of these four engines is already shown in

Figure 4.

For engine 53, the actual RUL is very close to the mean predicted RUL. The error is thus only -3.0 flights. Also $CRPS_{53}$, which is the generalization of the absolute error, is only 2.96. However, most of the mass of the predicted distribution of the RUL is on the right of the actual RUL, i.e., the RUL is overestimated (see Figure 4a). This is reflected in the relatively high $CRPS_{53}^W$ of 3.72. The actual RUL falls both within the $\alpha = 0.5$ and $\alpha = 0.95$ credible interval, and the widths of these intervals (15 and 45 flights respectively) are relatively small compared to engines 4, 86 and 67.

For engine 4, the mean predicted RUL is close to the actual RUL, with an error of 3.2 flights. Thus $CRPS_4$ is only 3.49. Also, $CRPS_4^W = 2.68$, which is less than $CRPS_4$. This is because most of the mass of the predicted pdf of the RUL is on the left of the actual RUL, i.e., the RUL is underestimated (see Figure 4b). The $\alpha = 0.5$ and $\alpha = 0.95$ credible interval both contain the actual RUL, but the width of these intervals (19 and 54 flights respectively) is relatively large compared to the other 3 engines. The low sharpness of this RUL distribution is thus reflected in the large widths of the credible intervals.

For engines 86 and 67, the mean predicted RUL is far off the actual RUL. This is reflected in the high CRPS values of 17.98 and 30.74, respectively. Moreover, nearly all the mass of the predicted pdf of the RUL of both engines is on the right of the actual RUL, i.e., the RUL is overestimated (see Figures 4c and 4d). The weighted CRPS metric is thus 26.96 and 46.11, respectively. This is higher than the standard CRPS metric for these two engines. The actual RUL of engine 86 falls within the $\alpha = 0.95$ credible interval, but the actual RUL of engine 67 does not.

5. CONCLUSIONS

In this paper, we have introduced novel metrics to evaluate the predicted probability distribution (pdf) of the RUL of components. The CRPS and $CRPS^W$ metrics evaluate the accuracy and sharpness of the estimated RUL distributions. The α -Coverage and Reliability Scores evaluate the reliability of the RUL prognostics.

We illustrate the four metrics for probabilistic RUL prognostics of the turbofan engines in the C-MAPSS dataset. We obtain these probabilistic RUL prognostics using a CNN with Monte Carlo dropout. The results show the distribution of the RUL of the turbofan engines is well estimated using this method. Moreover, the accuracy, sharpness and reliability of the obtained probabilistic RUL prognostics are shown to be well evaluated by our proposed metrics. Future studies that determine probabilistic RUL prognostics could therefore benefit from evaluating their results using these proposed metrics.

REFERENCES

- Babu, G. S., Zhao, P., & Li, X. L. (2016). Deep convolutional neural network based regression approach for estimation of Remaining Useful Life. In *International conference on database systems for advanced applications* (pp. 214–228).
- Baraldi, P., Mangili, F., & Zio, E. (2015). A prognostics approach to nuclear component degradation modeling based on gaussian process regression. *Progress in Nuclear Energy*, 78, 141–154.
- Biggio, L., Wieland, A., Chao, M. A., Kastanis, I., & Fink, O. (2021). Uncertainty-aware prognosis via deep gaussian process. *IEEE Access*, 9, 123517–123527.
- Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1), 1–3.
- de Pater, I., & Mitici, M. (2021). Predictive maintenance for multi-component systems of repairables with remaining-useful-life prognostics and a limited stock of spare components. *Reliability Engineering & System Safety*, 214, 107761.
- de Pater, I., Reijns, A., & Mitici, M. (2022). Alarm-based predictive maintenance scheduling for aircraft engines with imperfect remaining useful life prognostics. *Reliability Engineering & System Safety*, 108341.
- Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International conference on machine learning* (pp. 1050–1059).
- Gal, Y., Hron, J., & Kendall, A. (2017). Concrete dropout. *arXiv preprint arXiv:1705.07832*.
- Gneiting, T., & Katzfuss, M. (2014). Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1, 125–151.
- Gneiting, T., Raftery, A. E., Westveld III, A. H., & Goldman, T. (2005). Calibrated probabilistic forecasting using ensemble model output statistics and minimum crps estimation. *Monthly Weather Review*, 133(5), 1098–1118.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Lall, P., Lowe, R., & Goebel, K. (2011). Prognostics and health monitoring of electronic systems. In *2011 12th international conference on thermal, mechanical & multi-physics simulation and experiments in microelectronics and microsystems* (pp. 1–17).
- Lee, J., & Mitici, M. (2020). An integrated assessment of safety and efficiency of aircraft maintenance strategies using agent-based modelling and stochastic petri nets. *Reliability Engineering & System Safety*, 202, 107052.
- Lee, J., & Mitici, M. (2022). Multi-objective design of aircraft maintenance using gaussian process learning and adaptive sampling. *Reliability Engineering & System*

Safety, 218, 108123.

- Le Son, K., Fouladirad, M., & Barros, A. (2016). Remaining useful lifetime estimation and noisy gamma deterioration process. *Reliability Engineering & System Safety*, 149, 76–87.
- Li, X., Ding, Q., & Sun, J.-Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172, 1–11.
- Mitici, M., & de Pater, I. (2021). Online model-based remaining-useful-life prognostics for aircraft cooling units using time-warping degradation clustering. *Aerospace*, 8(6), 168.
- Nguyen, K. T., & Medjaher, K. (2019). A new dynamic predictive maintenance framework using deep learning for failure prognostics. *Reliability Engineering & System Safety*, 188, 251–262.
- Nowotarski, J., & Weron, R. (2018). Recent advances in electricity price forecasting: A review of probabilistic forecasting. *Renewable and Sustainable Energy Reviews*, 81, 1548–1568.
- Ramasso, E., & Saxena, A. (2014). Review and analysis of algorithmic approaches developed for prognostics on cmappss dataset. In *Annual conference of the prognostics and health management society 2014*.
- Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008). Metrics for evaluating performance of prognostic techniques. In *2008 international conference on prognostics and health management* (pp. 1–17).
- Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2009). Evaluating algorithm performance metrics tailored for prognostics. In *2009 IEEE aerospace conference* (pp. 1–13).
- Saxena, A., & Goebel, K. (2008). Turbofan engine degradation simulation data set. *NASA Ames Prognostics Data Repository* (<https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>), 878–887.
- Vandal, T., Livingston, M., Pihó, C., & Zimmerman, S. (2018). Prediction and uncertainty quantification of daily airport flight delays. In *International conference on predictive applications and APIs* (pp. 45–51).
- Xia, J., Feng, Y., Lu, C., Fei, C., & Xue, X. (2021). LSTM-based multi-layer self-attention method for Remaining Useful Life estimation of mechanical systems. *Engineering Failure Analysis*, 125, 105385.
- Zoutendijk, M., & Mitici, M. (2021). Probabilistic flight delay predictions using machine learning and applications to the flight-to-gate assignment problem. *Aerospace*, 8(6), 152.

BIOGRAPHIES

Ingeborg de Pater is a PhD candidate at the Faculty of Aerospace Engineering, Delft University of Technology, the Netherlands. Her research interests are predictive aircraft maintenance scheduling and Remaining-Useful-Life estimation of aircraft components.

Mihaela Mitici is an Assistant professor at the Faculty of Aerospace Engineering, Delft University of Technology. She has a PhD in Stochastic Operations Research, Department of Applied Mathematics, University of Twente, the Netherlands. Her research interests are stochastic operations research, stochastic processes and machine learning algorithms with applications to predictive aircraft maintenance scheduling and RUL prognostics.

APPENDIX: PYTHON CODE FOR THE NOVEL METRICS

```

#-----Necessary packages-----#
import numpy as np
import matplotlib.pyplot as plt
import sys

def CRPS(true_RULs, RUL_distributions, beta = 1.5):
    """
    This function calculates the CRPS and the weighted CRPS.
    Parameters
    -----
    true_RULs: Dictionary
        A dictionary with for each test instance (key, integer), the true RUL (value).
    RUL_distributions : Dictionary
        A dictionary with for each test instance (key, integer), a list (value) with all RUL predictions
        of this test instance. true_RULs and RUL distributions should have the same set of keys.
    beta : Float between 1 (included) and 2 (included)
        Penalty for overestimating the RUL relative to underestimating the RUL.
        The default is 1.5.
    Returns
    -----
    crps : Float
        The CRPS metric.
    weighted_crps : Float
        The weighted CRPS metric,
    """
    crps_sum = 0 #The value of the sum of the CRPS metric
    weighted_crps_sum = 0 #The value of the sum of the weighted CRPS metric

    #Calculate the CRPS and the weighted CRPS for each individual test instance
    for i in true_RULs.keys():
        #Initilize the CRPS and the weighted CRPS for test instance i
        crps_i = 0
        weighted_crps_i = 0

        #Get the probability distribution of the RUL of test instance i, and the true RUL
        distribution = RUL_distributions.get(i)
        true_RUL = true_RULs.get(i)
        distribution.sort()
        number_of_predictions = len(distribution) #The number of RUL predictions in the distribution

        for j in range(0, number_of_predictions -1, 1): #Go over all the predictions
            #Calculate the distance between two RUL predictions
            RUL_prediction = distribution[j]
            next_RUL_prediction = distribution[j+1]
            delta_RUL = next_RUL_prediction - RUL_prediction

            #Each RUL prediction has a probability of 1 over the number of predictions.
            #We use j+1, since j starts at 0, and since we consider the CDF
            probability = (j+1) / number_of_predictions

            #Check if the RUL prediction is larger, or smaller than the true RUL,
            #and update the CRPS and the weighted CRPS accordingly
            if RUL_prediction < true_RUL:
                probability_squared = probability ** 2
                crps_i = crps_i + (probability_squared * delta_RUL)
                weighted_crps_i = weighted_crps_i + (2 - beta) * (probability_squared * delta_RUL)
            else:
                probability_minus_one = probability - 1
                probability_squared = probability_minus_one ** 2
                crps_i = crps_i + (probability_squared * delta_RUL)
                weighted_crps_i = weighted_crps_i + beta * (probability_squared * delta_RUL)

        #Also consider the difference between the true RUL and the last prediction
        last_prediction = distribution[-1]
        if last_prediction < true_RUL:
            crps_i = crps_i + (1 * (true_RUL - last_prediction))
    
```

```

        weighted_crps_i = weighted_crps_i + (2 - beta) * (1 * (true_RUL - last_prediction))

    #Also consider the difference between the true RUL and the first prediction
    first_prediction = distribution[0]
    if first_prediction > true_RUL:
        crps_i = crps_i + (1 * (first_prediction - true_RUL))
        weighted_crps_i = weighted_crps_i + beta * (1 * (first_prediction - true_RUL))

    #Update the sum of the CRPS and the sum of the weighted CRPS
    crps_sum = crps_sum + crps_i
    weighted_crps_sum = weighted_crps_sum + weighted_crps_i
    #Take the average value of the CRPS and the weighted CRPS
    crps = crps_sum / len(RUL_distributions.keys())
    weighted_crps = weighted_crps_sum / len(RUL_distributions.keys())
    return crps, weighted_crps

def coverage(true_RULs, RUL_distributions, alpha):
    """
    This function computes the alpha-coverage and corresponding alpha-mean width.
    Parameters
    -----
    true_RULs: Dictionary
        A dictionary with for each test instance (key, integer), the true RUL (value).
    RUL_distributions : Dictionary
        A dictionary with for each test instance (key, integer), a list (value) with all RUL predictions
        of this test instance. true_RULs and RUL distributions should have the same set of keys.
    alpha : Float between 0 (included) and 1 (included)
        The desired width of the credible interval.
    Returns
    -----
    coverage : Float between 0 (included) and 1 (included)
        The coverage belonging to alpha.
    mean_width : Float
        The mean width of the credible interval belonging to alpha.
    """
    #Initialize the parameters of the credible interval
    total_width = 0 #Total width of all credible intervals
    in_ci = 0 #The number of components for which the true RUL falls within the credible interval
    percentile_lower = 0.5 - 0.5 * alpha #Lower percentile of the credible interval
    percentile_higher = 0.5 + 0.5 * alpha #Upper percentile of the credible interval

    #Check for each test instance i if the true RUL falls inside,
    #or outside the credible interval of test instance i
    for i in true_RULs.keys():
        #Get the probability distributions of the RUL test instance i, and the true RUL
        distribution = RUL_distributions.get(i)
        true_RUL = true_RULs.get(i)
        distribution.sort()
        number_of_predictions = len(distribution) #The number of RUL predictions in the distribution

        #Get the indexes of the RUL predictions belonging to the considered percentiles.
        #We use -1, since a list in python starts at 0 instead of 1
        index_lower = max(0, int(percentile_lower * number_of_predictions) - 1)
        index_higher = int(percentile_higher * number_of_predictions) - 1
        lower_bound_ci = distribution[index_lower] #Lower bound credible interval
        upper_bound_ci = distribution[index_higher] #Upper bound credible interval

        #Check if the true RUL is within the credible interval
        if true_RUL >= lower_bound_ci and true_RUL <= upper_bound_ci:
            in_ci = in_ci + 1

        #Update the total width of all credible interval
        total_width = total_width + (upper_bound_ci - lower_bound_ci)
    #Calculate the coverage and the mean width of the credible interval
    coverage = in_ci / len(true_RULs.keys())
    mean_width = total_width / len(true_RULs.keys())
    return coverage, mean_width

```

```

def area_under(x_1, x_2, f_1, f_2):
    """
    This functions calculates the area between the ideal curve and the reliability curve, between
    x_1 and x_2. Here, the reliability curve is under the ideal curve between x_1 and x_2.
    Parameters
    -----
    x_1 : Float
        Start value of alpha.
    x_2 : Float
        End value of alpha.
    f_1 : Float
        Coverage at alpha = x_1.
    f_2 : Float
        Coverage at alpha = x_2.
    Returns
    -----
    area : Float
        Area between the ideal curve and the reliability curve, between x_1 and x_2.
    """
    area = (x_2 - f_2) * (x_2 - x_1) - 0.5 * (x_2 - x_1) * (x_2 - x_1)
    area = area + 0.5 * (x_2 - x_1) * (f_2 - f_1)
    return area

def area_above(x_1, x_2, f_1, f_2):
    """
    This functions calculates the area between the ideal curve and the reliability curve, between
    x_1 and x_2. Here, the reliability curve is above the ideal curve between x_1 and x_2.
    Parameters
    -----
    x_1 : Float
        Start value of alpha.
    x_2 : Float
        End value of alpha.
    f_1 : Float
        Coverage at alpha = x_1.
    f_2 : Float
        Coverage at alpha = x_2 .
    Returns
    -----
    area : Float
        Area between the ideal curve and the reliability curve, between x_1 and x_2.
    """
    area = (f_1 - x_1) * (x_2 - x_1) - 0.5 * (x_2 - x_1) * (x_2 - x_1)
    area = area + 0.5 * (x_2 - x_1) * (f_2 - f_1)
    return area

def reliability_score(true_RULs, RUL_distributions, name, stepsize = 0.01 ):
    """
    This functions calculates the reliability scores (under, over and total),
    and plots the reliability diagram.
    Parameters
    -----
    true_RULs: Dictionary
        A dictionary with for each test instance (key, integer), the true RUL (value).
    RUL_distributions : Dictionary
        A dictionary with for each test instance (key, integer), a list (value) with all RUL predictions
        of this test instance. true_RULs and RUL distributions should have the same set of keys.
    Returns
    -----
    RS_total : Float
        Total Reliability Score.
    RS_under : Float
        Reliability Score - underestimation of uncertainty.
    RS_over : Float
        Reliability Score - overestimation of uncertainty.
    """

```

```

#-----Calculate the Reliability curve
reliability_curve = []
for alpha in np.arange(0, 1 + sys.float_info.epsilon, stepsize): #One is included
    alpha_coverage = coverage(true_RULs, RUL_distributions, alpha)[0]
    reliability_curve.append(alpha_coverage)

#-----Plot the reliability diagram
ideal_curve = list(np.arange(0, 1 + sys.float_info.epsilon, stepsize)) #ideal curve, where y = x
fig, ax = plt.subplots()
ax.set_ylabel(r"$\alpha\mathrm{-Coverage}$", fontsize = 16)
ax.set_xlabel(r"$\mathrm{Width \: of \: credible \: interval \: } \alpha$", fontsize = 16)
ax.plot(ideal_curve, ideal_curve, label = "Ideal curve", c = "red")
ax.plot(ideal_curve, reliability_curve , label = "Reliability curve", color = "blue", \
        linestyle = "dashed")
ax.legend()
plt.show()

#-----Calculate the reliability score
RS_under = 0 #The reliability score: underestimation of the uncertainty
RS_over = 0 #The reliability score: overestimation of the uncertainty

for alpha in np.arange(0, 1, stepsize): #Loop over all alpha's
    next_alpha = alpha + stepsize
    coverage_alpha = coverage(true_RULs, RUL_distributions, alpha)[0]
    coverage_next_alpha = coverage(true_RULs, RUL_distributions, next_alpha)[0]

    #If the reliability curve is beneath the ideal curve:
    if coverage_alpha <= alpha and coverage_next_alpha <= next_alpha:
        surface = area_under(alpha, next_alpha, coverage_alpha, coverage_next_alpha)
        RS_under = RS_under + surface

    #If the reliability curve is above the ideal curve:
    elif coverage_alpha >= alpha and coverage_next_alpha >= next_alpha:
        surface = area_above(alpha, next_alpha, coverage_alpha, coverage_next_alpha)
        RS_over = RS_over + surface

    #If the reliability curve starts under the ideal curve, and ends above the ideal curve
    elif coverage_alpha <= alpha and coverage_next_alpha >= next_alpha:
        #Find the place where the reliability curve crosses the ideal curve
        dy = coverage_next_alpha - coverage_alpha
        a = dy / stepsize
        alpha_cross = (coverage_alpha - a * alpha) / (1-a)
        coverage_cross = alpha_cross

        #Calculate the surface under the ideal curve
        surface_under = area_under(alpha, alpha_cross, coverage_alpha, coverage_cross)
        RS_under = RS_under + surface_under
        #Calculate the surface above the ideal curve
        surface_above = area_above(alpha_cross, next_alpha, coverage_cross, coverage_next_alpha)
        RS_over = RS_over + surface_above

    #If the reliability curve starts above the ideal curve, and ends under the ideal curve
    elif coverage_alpha >= alpha and coverage_next_alpha <= next_alpha:
        #Find the place where the reliability curve crosses the ideal curve
        dy = coverage_next_alpha - coverage_alpha
        a = dy / stepsize
        alpha_cross = (coverage_alpha - a * alpha) / (1-a)
        coverage_cross = alpha_cross

        #Calculate the surface above the ideal curve
        surface_above = area_above(alpha, alpha_cross, coverage_alpha, coverage_cross)
        RS_over = RS_over + surface_above
        #Calculate the surface under the ideal curve
        surface_under = area_under(alpha_cross, next_alpha, coverage_cross, coverage_next_alpha)
        RS_under = RS_under + surface_under
RS_total = RS_under + RS_over #Total reliability score
return RS_total, RS_under, RS_over

```

Filtering Misleading Repair Log Labels to Improve Predictive Maintenance Models

Pablo del Moral¹, Sławomir Nowaczyk¹, and Sepideh Pashami^{1,2}

¹ *Center for Applied Intelligent Systems Research CAISR, Halmstad University*

pablo.del_moral@hh.se

slawomir.nowaczyk@hh.se

sepideh.pashami@hh.se

² *RISE Research Institutes of Sweden*

sepideh.pashami@ri.se

ABSTRACT

One of the main challenges for predictive maintenance in real applications is the quality of the data, especially the labels. In this paper, we propose a methodology to filter out the misleading labels that harm the performance of Machine Learning models. Ideally, predictive maintenance would be based on the information of when a fault has occurred in a machine and what specific type of fault it was. Then, we could train machine learning models to identify the symptoms of such fault before it leads to a breakdown. However, in many industrial applications, this information is not available. Instead, we approximate it using a log of component replacements, usually coming from the sales or maintenance departments. The repair history provides reliable labels for fault prediction models only if the replaced component was indeed faulty, with symptoms captured by collected data, and it was going to lead to a breakdown.

However, very often, at least for complex equipment, this assumption does not hold. Models trained using unreliable labels will then, necessarily, fail. We demonstrate that filtering misleading labels leads to improved results. Our central claim is that the same fault, happening several times, should have similar symptoms in the data; thus, we can train a model to predict them. On the contrary, replacements of the same component that do not exhibit similar symptoms will be confusing and harm the ML models. Therefore, we aim to filter the maintenance operations, keeping only those that can be used to predict each other. Suppose we can train a successful model using the data before a component replacement to predict another component replacement. In that case, those maintenance operations must be motivated by the same, or a

very similar, type of fault.

We test this approach on a real scenario using data from a fleet of sterilizers deployed in hospitals. The data includes sensor readings from the machines describing their operations and the service logs indicating the replacement of components when the manufacturing company performs the service. Since sterilizers are complex machines consisting of many components and systems interacting with each other, there is the possibility of faults happening simultaneously.

1. INTRODUCTION

In this paper, we are going to deal with the common industrial problem of learning predictive maintenance models using labels from repair logs. This work has been inspired by the collaboration with our industrial partner Getinge AB, a company producing sterilizers to be used at hospitals for medical equipment. The machine sterilizes its load by using phases of low pressure to eliminate air and humidity combined with phases of high temperature that kill micro-organisms. A sterilizer is critical for the hospital's operation: without properly sterilized material, most daily activities can not be executed. Getinge also provides service and maintenance. Unexpected failures often lead to long downtimes: a technician needs to be sent to the machine, the problem diagnosed, the right parts ordered and installed, and the machine needs to be tested.

The ideal scenario to train a predictive model is to monitor a machine while undergoing a fault, record the data describing its operation in the meantime, and find patterns that describe the effect or symptoms of this fault. Later, these patterns can be used to identify similar faults in the future so that maintenance actions can be performed before the issue leads to an unexpected breakdown. This avoids consequential or collateral costs associated with a breakdown without the increase in maintenance costs associated with preventive maintenance

Pablo Del Moral et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

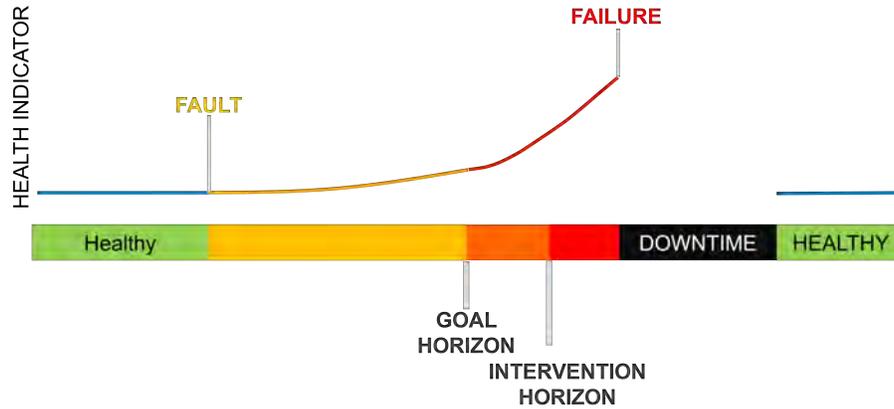


Figure 1. An idealized overview of the setup and the data needed to train a predictive maintenance model for estimating Remaining Useful Life.

schemes.

This ideal scenario is possible to realize during the design of the machine but relatively difficult during real-world operation. Analyzing different faults under controlled conditions can be done in a lab setting while designing and deciding what data to record to monitor the evolution of the fault properly. However, in our application scenario and many similar ones, this ideal scenario is not feasible. First, tests under controlled conditions do not necessarily reflect the real operation of the machines in the field. External conditions such as water temperature, humidity, load, and usage considerably affect the functioning of the machine. Reconstructing all of them in the lab is a daunting task, not viable economically. Even more importantly, our industrial partners need to develop predictive maintenance models on already built and deployed machines. Like many other machines, sterilizers are expected to work for many years, often decades; and it is precisely those older machines that have the room for the greatest gains from a new, ML-based, successful predictive maintenance solution.

In other words, we would like to learn predictive maintenance models using data from the machines that are already deployed in the field. These machines have been collecting sensor data for several years, mainly for purposes of control and security. We can use this data to find patterns to predict future failures – however, one has to keep in mind that the data is far from ideal since it has generally not been collected for the purpose of monitoring the health of the machines.

Another aspect to consider is that we do not have direct, reliable information about the faults that happened in the machines. However, we need at least some approximation of a history of past faults to be able to label the sensor data and build predictive models. In reality, the best information that we can typically get is the maintenance logs or repair histories, i.e., a register about when maintenance was done and

which components were installed in each machine. This way of labeling the data is very unreliable for various reasons that we will present in detail in the next section; if used naively, it necessarily leads to inaccurate models.

In this paper, we present a methodology to filter these repair logs, selecting only the labels that can be used to provide reliable predictions. The key concept is based on the following observation. If the same *predictable* fault happens in several machines, similar patterns will be found in the sensor data. These faults will develop into failures; then maintenance will be carried out, and components will be replaced; we will obtain information about those component replacements (only, not of the faults directly) through the maintenance logs. Our underlying assumption is that the inverse should also hold. Suppose we can identify patterns in the data before a component replacement that are useful to predict other component replacements. In that case, those component replacements must be related to a similar type of fault. This approach, on its own, would be very prone to overfitting; therefore, we add a second step to our method, where we refine our models by adding the false alarms created in the first step.

The rest of the paper is structured as follows: in Section 2, we motivate our research, specifically focusing on why the service logs are often unreliable; in Section 3, we present the different steps of our approach; in Section 4, we summarize our experiments and discuss the results; in Section 5, we compare our work against state of the art and provide the literature review; and in Section 6, we present our conclusions and future work.

2. PROBLEM FORMULATION

This section will describe some of the problems typically encountered while labelling predictive maintenance data using information coming from service logs.

Before we focus on the labels, though, it is important to note that the data recorded in the machines is generally not designed to describe the health of the machine and its components accurately. Therefore, it is expected that many faults will show no symptoms in the data or will show confusing symptoms. Our task is to find as many patterns as possible in this data that could be linked to faults that later lead to failures.

In Figure 1, we have an example of the idealized training case to create models that predict faults. The machine is in a healthy state until a fault happens. We are recording a signal (one or multi-variate) that perfectly describes the machine's state of health. Once the fault is introduced, this health begins to deteriorate, and the progression is reflected in the selected health indicator. At some time, it reaches the failure point, and the machine stops working. After the failure, a repair is needed, which means that the machine will be out of operation until it is repaired. Then, the machine goes back to operation at full health.

Planning and executing a maintenance operation requires time. We define this time as the "intervention horizon": the last moment before a failure when it is possible to intervene on the machine and avoid it. The goal horizon adds a safety margin to schedule the maintenance operation (with perfect models, the goal horizon and the intervention horizon would be the same). We are recording data to obtain a health indicator; we can label this data and train a classifier with it based on the goal horizon. The data before the goal horizon would be "SAFE," and the data after the goal horizon would be "ALARM." In the future, when the model predicts an "ALARM," we can schedule a maintenance operation in the most convenient way and avert the failure.

However, in real-life applications, when we must rely on using maintenance records, this ideal case does not happen. The only information we have is when a component was replaced, in a given machine. This presents a number of ambiguities:

- There are uncertainties in the dates. Even assuming that the dates are entered correctly (which is not always the case due to human error or accounting policies), we know when a component was replaced in the machine. However, we do not know how long it took from the failure to the component replacement.
- In fact, we do not know if the replaced component had failed or not. The replacement of a component can be due to a failure, but it can also be due to a preventive maintenance operation or a subjective decision by the technician.
- If there was a fault happening in the machine, we do not know if the replacement of the component solved it. In other words, we can not be sure that the diagnosis by the technician doing maintenance was correct.

- If many components were replaced, we do not know which one was responsible for the fault or if different faults were happening simultaneously. Usually, multiple symptom patterns can be identified in the data – and matching them to replaced components is error-prone.
- We do not have the certainty that all the maintenance operations are recorded in the service logs. Some of the maintenance operations can be carried out by the staff operating the machines. In addition, hospitals can buy service from different companies.
- In the case where the failure actually happened and the responsible component was replaced, we do not have information about when the fault started.

These uncertainties will lead to wrong labeling of the sensor data recorded. The effect of such wrong labeling is particularly harmful because we are not just labeling one data point but a full sequence of data, from the moment the fault starts, until the failure happens.

To sum up, using the maintenance records, the only certainties we can obtain are the intervention and goal horizons. We can use the goal horizon to label the recorded data by the machine as "ALARM" until the component is replaced. We can label an arbitrary amount of the data before the goal horizon as "SAFE."

3. LITERATURE REVIEW

Predictive maintenance is a hot topic in the research and industrial communities. According to (?), maintenance strategies based on corrective resulted in more than 3 times more downtime and 16 times more defects than more advanced maintenance strategies.

A quick overview of recent surveys (?), (?), shows how most data-driven methods for predictive maintenance use supervised methods, i.e., reliable information about the historical faults are needed to train models for predictive maintenance.

There are different approaches to obtaining accurate labels. One solution is to use simulated data, where the operation of a machine or system is simulated, and faults are introduced at known times. Examples of these datasets are the "Turbofan Engine Degradation Simulation Data Set" (?), or the "Tennessee Eastman Process" (?).

Another option is to run tests done in a laboratory (?). In the field of fault prediction for bearing machinery, this approach is very popular: a setup is built, data is recorded, and faults are introduced. Again, the moment when the fault was introduced is clearly determined, and the evolution of the fault is carefully monitored.

The problem of using repair logs as a reliable source of information has also been researched. In (?), the authors present

an approach to complete the information of the repair logs using Natural Language Processing approaches to determine which component was the recipient of a particular maintenance operation.

In (? , ?), the authors describe some of the uncertainties coming from using the information from the repair logs as a source of labels for the data recorded in trucks. Although not focusing directly on this problem, they study how uncertainties in the dates can have a significant role in setting parameters such as the prediction horizon.

Finally, in (? , ?), the authors use log data from medical equipment to predict future failures. They discuss the problems of relying on the information coming from the repair logs and build their approach to solve this problem. The main difference between their problem definition and ours is that they can assume that the absence of repair logs for a given time means that the machine is healthy, while we can not make this assumption in our practical case.

From the technical point of view of machine learning, our scenario is related to the task of learning in the presence of noisy labels. According to the taxonomies presented in (? , ?), we can categorize our approach as "model predictions-based filtering" since we use the performance of a classifier as a tool to filter the label noise. As an example, in (? , ?), the authors use the output of a neural network during the training to detect the noisy labels and filter out the corresponding instances. The main difference between our approach and the state-of-te-art stems from the nature of the noise in our data: the uncertainty resides in the repair logs, which are used not just to label a single instance, but a complete sequence of instances.

4. METHOD

4.1. Data

For this study, we work with the data coming from 67 machines situated in several different countries. The data coming from the machines contain sensor data measuring magnitudes such as pressure or temperature inside the chamber of the sterilizer during its processes. Although all the sterilizers are part of the same product line, there exist different models with, for example, different sizes of the sterilization chamber or different versions of particular components.

We will use two years of recorded data produced by the machines. Not all the machines have produced data for the whole period, and most of them have been deployed on the field much longer than these two years. For each process of the machine (usually called cycle), the raw data contains sensor reading for the pressure and the temperature in different parts of the chamber. Working with the experts, we have extracted 86 features that characterize each cycle. Typically, a machine runs 5-10 cycles per day. Per regulation, one of these cy-

NUMBER OF MAINTENANCE OPERATIONS PER MACHINE

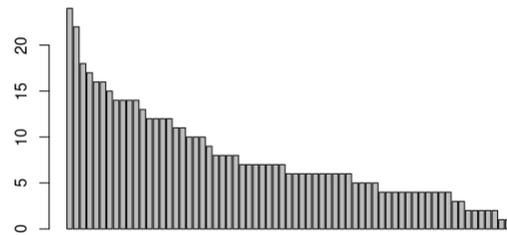


Figure 2. Number of maintenance operations per machine.

cles has to be run on an empty load to validate with certain biomarkers that the machine is still achieving sufficient sterilization.

For those two years of data, a total of 275 maintenance events with component replacements have happened. In Figure 2, we can see the distribution of the number of maintenance events with component replacement per machine. Some machines have very few component replacements logged in the service logs, while others have a larger number.

4.1.1. Data Preprocessing

Every machine has a slightly different configuration and can be used in different settings. For example, different machines can have different sizes of the chamber, which obviously affects how the extracted features from the data look like. But even for the same machine, conditions can change over its lifetime, for example, by upgrading one of the components or due to changes in the usage patterns.

After discussion with the experts, two factors need to be considered. First, the data is subject to small random variations based on factors like the room's water temperature and humidity, among others. Second, to normalize the data across machines, we should focus on the rate of change of the features, not on their absolute values.

With this in mind, we first calculate a moving average of all signals to reduce the effect of small random variations in the data. For every cycle, the value for each feature is the average over the previous 100 cycles. Then, we take the difference between the value of the moving average of the current cycle and the cycle 100 cycles before. Thus, the final value for the features of a given cycle is calculated by considering the previous 200 cycles.

4.2. Filtering of Component Replacements

Based on the discussion in Section 2, we label the data 75 cycles before a component replacement as "ALARM" and

the data between 150 and 75 cycles before the component replacement as "SAFE." This corresponds approximately to 15 and 30 days before the component replacement. This is in line with the time it would take to schedule and perform maintenance on the machine.

Our assumption is that if we can train a classifier on the data before a component replacement, and use that classifier to accurately predict another component replacement, then those two component replacements are most likely related to the same fault.

Algorithm 1 Fitness function.

```

1:  $N$  set of events
2:  $D_i = [(\mathbf{x}^1, y^1) \dots (\mathbf{x}^m, y^m)]$  dataset for event  $i$ 
3:  $R$  vector of results
4: for  $i = 1$  to  $length(N)$  do
5:    $D_{train} = \bigcup_{j \neq i} D_j$ 
6:    $m = \text{trainclassifier}(D)$ 
7:    $\text{pred} = \text{predictprobability}(D_i)$ 
8:    $R_i = \text{measureAUC}(\text{pred}, \mathbf{y}_i)$ 
9: end for
10:  $F$ : fitness value
11: for  $r = 1$  to  $length(N)$  do
12:   if  $R_i > \text{threshold}$  then
13:      $F = F + 1$ 
14:   else
15:      $F = F - 1/3$ 
16:   end if
17: end for

```

With a number N of component replacement events available, our goal is to select the largest subgroup of events that can be used to train classifiers that can predict each other. Each event i is associated with a training set $D_i = [(\mathbf{x}^1, y^1) \dots (\mathbf{x}^m, y^m)]$, where \mathbf{x}^1 is the feature vector, and y is the associated label ("SAFE" or "ALARM"). The number of possible subgroups increases dramatically as we consider more and more component replacement events. To perform this search, we use a genetic algorithm. In Algorithm 1, we present the fitness function used. The goal is to select as many events as possible such that they can all be used to predict each other, while discarding as many events as possible that can not be predicted.

To implement the genetic algorithm, we use the *ga* function from the **GA** package in R. The population at each generation is 50, the probability of cross-over is 0.8, the probability of mutation is 0.1, and the elitism is set at 0.1. We choose to stop the search after 10 iterations without improvements in the best solution.

4.3. Experimental Setup and Evaluation.

We split the two years of data into four periods of half a year each. For each period, we train our models with all the recorded data and component replacement events until that period. Then, we evaluate in the following period(s).

The evaluation is based on the ability to predict a future component replacement. Since there is a certain degree of randomness in the real data, we wait for three predictions of "ALARM" before actually issuing an alarm and dispatching the technician. If a component replacement happens in the following 75 cycles, it is marked as a correct prediction. It will be considered an early alarm if a component replacement happens between 75 and 100 cycles. If a component replacement is performed without a previous alarm, it is a missed failure. Finally, if no component replacement is performed in the next 100 cycles, it is a false alarm. In a realistic scenario, when an alarm is issued, the technician is sent to check the status of the machine. If the technician finds that the machine is in a healthy state, we could consider the alarm to be false and the machine to be healthy for the near future. However, the models would likely keep issuing alarms, that could be ignored based on the expertise of the technician. For this reason, in our evaluation, we observe a cooldown period of 25 cycles after a false alarm.

4.4. Refinement of Component Selection

In the recorded sensor data, we can often observe many trends that turn out to be unrelated to the presence or absence of faults. For example, these trends typically relate to the external weather conditions, the temperature of the water, or the usage of the machines. The component replacement event selection from Subsection 4.2 can be very prone to picking up these spurious trends; since the number of examples is very low, it will therefore commonly lead to overfitting.

In order to avoid overfitting, we propose to add these false alarms as "soft labels" into the training process. The process is described in Figure 3. If we train a model with the data for some time, we will evaluate and identify the false alarms in the following testing period. For each of these false alarms, we will extract a data sample with the previous 150 cycles and add them to the selection process of Subsection 4.2. These data samples can be added to the training datasets, labeled as "SAFE." In practice, this means training models that try to predict as many faults as possible while keeping the number of false alarms low.

On the other hand, looking at Figure 2, it is natural to think that the difference in the number of maintenance operations between machines is not necessarily due to some machines being inherently better than the others. In other words, there is probably missing information in the maintenance history of some machines. This means that an unknown number of false alarms should be expected; that, in fact, they are not false alarms, and a fault might be happening. If we were to introduce the data samples for these alarms in the training process, we would be introducing confusing labeling again. Therefore, we need to select those false alarms' data samples.

We again use a genetic algorithm to select the component re-

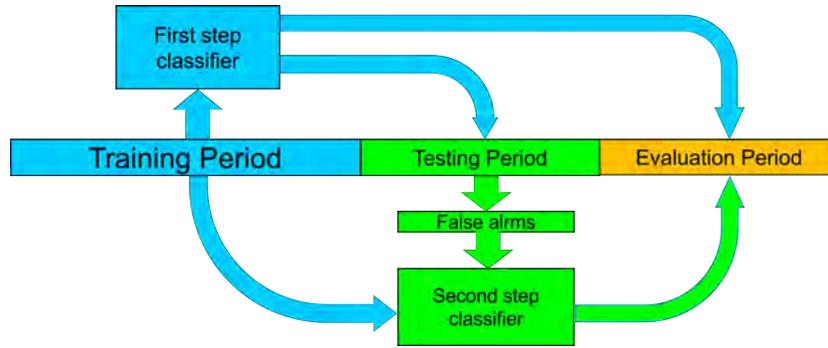


Figure 3. Workflow of the proposed methodology.

placement events that are useful to predict each other. In addition, we add the false alarms events that are not predicted as "ALARM."

Algorithm 2 Fitness function for the refinement step.

```

1:  $N$  set of component replacement events and false alarm
   events.
2:  $D_i = [(x^1, y^1) \dots (x^m, y^m)]$  dataset for event  $i$ 
3:  $R$  vector of results
4: for  $i = 1$  to  $length(N)$  do
5:    $D_{train} = \bigcup_{j \neq i} D_j$ 
6:    $m = \text{trainclassifier}(D)$ 
7:   pred =  $\text{predictprobability}(D_i)$ 
8:    $R_i = \text{measureAUC}(\text{pred}, y_i)$ 
9: end for
10:  $F$  fitness value
11: for  $r = 1$  to  $length(N)$  do
12:   if  $i$  corresponds to a component event then
13:     if  $R_i > \text{threshold}_1$  then
14:        $F = F + 1$ 
15:     else
16:        $F = F - 1/3$ 
17:     end if
18:   else
19:     if  $R_i < \text{threshold}_2$  then
20:        $F = F + 1/10$ 
21:     end if
22:   end if
23: end for

```

Now the fitness function in Algorithm 2 accounts for the number of component replacements that are correctly predicted. There is also a bonus for the number of false alarm events that are not predicted as "ALARM."

5. EXPERIMENTS AND RESULTS

Our goal in the following experiments is to compare the performance of the classifier trained on the selected component replacement events as described in Subsection 4.2 (from now on, First Step Model) and the performance of the classifier trained using the selected component replacement events and the false alarms created by the first model, as described in Subsection 4.4 (from now on, Second Step Model).

To do so, we train a First Step Model for a given period n . We use the period $n + 1$, to evaluate the presence of false alarms. We then use those false alarms to train a Second Step Model on the component replacement events of period n , and the false alarms created during period $n + 1$. We will compare both models on period $n + 2$ and the following. In practice, we have two years of data and four periods, so this means two comparisons.

5.1. Training During Period 1

In total, there are 34 component replacements in period 1. To select the fault component replacement events, we choose a threshold for the area under the ROC curve of 0.9, a very restrictive value.

Table 1. Results of predicting failures during periods 3 & 4, based on classifiers trained on period 1 (First Step) or periods 1+2 (Second Step). Comparison based on the Correctly Predicted Replaced Components (CPC), Early Alarms (EA), Missed Component Replacement (MCR) and False Alarms (FA).

	CPRC	EA	MCR	FA
First Step	24	0	70	76
Second Step	30	0	64	67

In selecting the component replacement events useful to predict each other, the genetic algorithm chooses 17 component replacement events, among which only 7 are predicted with more than 0.9 of area under the ROC curve. This means that only about 20% of the component replacement events are selected.

For the Second Step Classifier, we use the 34 component replacements during period 1, and the 76 false alarms from period 2. After the genetic algorithm performs the selection, 18 component replacement events were selected, among which only 5 had a predicted area under the ROC curve bigger than 0.9. In addition, 25 false alarms were selected.

The results of both approaches evaluated on periods 3 and 4 can be seen in Table 1. Not only is the Second Step reducing

the number of false alarms by more than 10% (keep in mind that an unknown number of false alarms is to be expected); but we have also increased the number of correctly predicted component replacements by 25%.

There are many missed component replacements. As an indication, though, we should keep in mind that we used 20-25% of the component replacement events for training. This value roughly coincides with the ratio of correctly predicted component replacement to missed component replacements.

5.2. Training During Periods 1 and 2.

In total, there are 98 component replacements during periods 1 and 2. To select the fault component replacement events, we use again the threshold for the area under the ROC curve of 0.9.

Table 2. Results of predicting failures during period 4, based on classifiers trained on periods 1+2 (First Step) or periods 1+2+3 (Second Step). Comparison based on the Correctly Predicted Replaced Components (CPRC), Early Alarms (EA), Missed Component Replacement (MCR) and False Alarms (FA).

	CPRC	EA	MCR	FA
First Step	7	0	40	53
Second Step	12	0	35	47

After selecting the component replacement events that are useful to predict each other, the genetic algorithm selects 53 component replacement events, among which only 26 are predicted with more than 0.9 of area under the ROC curve. This means that only about 25% of the component replacement events are selected.

For the Second Step Classifier, we use the 98 component replacements during periods 1 & 2 and the 53 false alarms from period 3. After the selection performed by the genetic algorithms, 51 component replacement events were selected, among which only 21 had a predicted area under the ROC curve bigger than 0.9. In addition, 11 false alarms were selected.

The results of both approaches evaluated on period 4 can be seen in Table 2. Like in the previous experiment, the two-step approach has significantly increased the number of correctly predicted component replacements and the final number of false alarms has been decreased.

The ratio of correctly predicted component replacements and missed component replacements is just 15%, compared to the 25% of component replacement events selected in the training phase. For the second step classifier, this ratio is about 20% in the selection phase and 25% in the evaluation phase.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a methodology to deal with the problem of misleading repair logs that can be harmful when creating machine learning-based predictive models. Using this misleading information to train models for predictive maintenance often leads to poor performance in practical settings when the quality of available data is not very high.

There are multiple reasons for this misleading information existing in reality. We can summarize them as follows: we cannot be sure that the replacement of a component in a machine is caused by the presence of a fault in said component; even if it is, we cannot be sure that reliable symptoms in the data exist to track the health deterioration; finally, we cannot be sure that no other maintenance operations have been performed in the machine, without being recorded in the service logs.

To deal with this problem, we first present a methodology to select those component replacements that are useful to create good performance models. This somehow naive selection process has been demonstrated experimentally to necessarily lead to overfitting and a large number of false alarms when those models are used to predict future failures.

We further add a second step, proposing our new methodology, where false alarms created in the first step are used to refine our models. We expected to reduce the number of false alarms after the refinement phase, which we have achieved by a margin of more than 10%, as verified experimentally on real-world industrial data.

However, more interestingly, we have also improved the number of correctly predicted component replacements by a healthy margin of 25% or more. Adding the false alarms to the training phase not only reduces the number of predicted false alarms in the future but also improves the selection of component replacement events to create more accurate models.

Implicitly, we have assumed that there is just one fault mode in our machines. By simply selecting the component replacement events that are useful to predict each other, we expect our selection process to just focus on one type of fault. In reality, we know that a machine as complex as a sterilizer will have many different types of faults. A clear continuation of this work is to extend the methodology to accommodate different types of faults either by adapting the selection algorithm to naturally accommodate them or by performing the selection iteratively.

ACKNOWLEDGMENT

This work was partially supported by "Stiftelsen för kunskaps- och kompetensutveckling" and CHIST-ERA grant CHIST-ERA-19-XAI-012 funded by Swedish Research Council.

BIOGRAPHIES

Pablo del Moral is a PhD candidate in Data Mining at Center for Applied Intelligent Systems Research, Halmstad University, Sweden. He has a Masters degree in Data Science from University of Granada, and a Masters degree in Nuclear, Particle and Astrophysics from Technical University of Munich.

Slawomir Nowaczyk is a Professor in Machine Learning, working at Center for Applied Intelligent Systems Research, Halmstad University, Sweden. He has received his MSc degree from Poznan University of Technology in 2002 and his PhD degree from Lund University of Technology in 2008. During the last decade his research focused on knowledge representation, data mining and self-organising systems, especially in large and distributed data streams, including unsupervised modelling. He is a board member for the Swedish AI Society, and a research leader for the School of Information Technology at the University of Halmstad. Slawomir has led multiple research projects related to applying Artificial Intel-

ligence and Machine

Learning in many different domains, such as transport and automotive, energy, smart cities as well as healthcare. In most cases, this research was done in collaboration with industry and public administration organisations – inspired by practical challenges and leading to tangible results and deployed solutions.

Sepideh Pashami is a senior researcher at RISE and a lecturer at CAISR (Center for Applied Intelligent Systems Research) at Halmstad University. She received her Ph.D. degree from AASS Research Centre, Örebro University, Sweden, in 2016. Her research interests include predictive maintenance, interactive machine learning, causal inference, and representation learning. She has been involved as a researcher and research leader in many projects (e.g. EVE, In4Uptime, ARISE and HEALTH) together with Volvo Group AB, applying machine learning for predictive maintenance of heavy-duty vehicles.

Physics-informed lightweight Temporal Convolution Networks for fault prognostics associated to bearing stiffness degradation

Weikun Deng¹, Khanh T. P. Nguyen², Christian Gogu², Jérôme Morio², and Kamal Medjaher³

^{1,2,3} *Laboratoire génie de production, Université de Toulouse, INP-ENIT, 47 Av. d' Azereix, 65000 Tarbes, France
weikun.deng@enit.fr, tnguyen@enit.fr, kamal.medjaher@enit.fr*

² *Institut Clément Ader, Université de Toulouse, 3 rue Caroline Aigle, 31400 Toulouse, France
christian.gogu@gmail.com*

² *ONERA/DTIS, Université de Toulouse, F-31055 Toulouse, France
Jerome.Morio@onera.fr*

ABSTRACT

This paper proposes hybrid methods using physics-informed (PI) lightweight Temporal Convolution Neural Network (PITCN) for bearings' remaining useful life (RUL) prediction under stiffness degradation. It includes three PI hybrid models: a) PI Feature model (PIFM) — constructing physics-informed health indicator (PIHI) to augment the feature space, b) PI Layer model (PILM) — encoding the physics governing equations in a hidden layer, and c) PI Layer Based Loss model (PILLM) — designing PI conflict loss, taking into account the difference before and after integration of the physics input-output relations involved module to the loss function. We simulated 200 different bearing stiffness degradations, using their discrete monitored vibration signals to verify the effectiveness of the proposed method. We also investigate their inference process through feature heat map analysis to interpret how the models melt physics knowledge to assist in capturing the degradation trend. The physics knowledge considered in this paper is the dynamic relationship between vibration amplitude and stiffness in a damped forced vibration model. The results show that all three PITCN models effectively capture degradation-related trend information and perform better than the vanilla lightweight TCN. Furthermore, the visualization of the feature channels highlights the important role of physics information in model training. Channels containing physics information demonstrate higher correlation with results as they significantly dominate the heat map compared to other channels.

Keywords—Physics-informed machine learning; Non-trending vibration; Prognostic; Bearing contact stiffness degradation

Weikun Deng et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

Bearings are critical components that are susceptible to fatigue damage and need to be monitored. Under certain degraded working conditions, its crack propagation process is not obviously characterized in the early stage monitoring. (Li, Hu, Meng, Zhan, & Shen, 2018). As a result, the monitoring signal and the corresponding periodical statistics tend to show “slight trend” before severe degradation, and the dramatic feature variation only appears in signal collected from the end of services life (Porotsky & Bluvband, 2012).

Due to the incomplete knowledge of the bearing degradation mechanism, high cost in dynamics failure modeling (Massi et al., 2014) and the sparsity of the degradation information, the classic solutions, neither the traditional physics-based methods nor the data-driven machine learning (ML) methods (Shi & Chehade, 2021) are applicable to capture the information about the nonlinear degradation from past data and working conditions (H. Liu, Song, Zhang, & Kudreyko, 2021).

As a result, researchers have turned to the quest to develop hybrid approaches. On the one hand, guiding machine learning to explore the embedding properties of the data by imposing additional constraints during training has been shown to learn better representations of degradation trends (Liao, Jin, & Pavel, 2016). On the other hand, the physics-informed Machine Learning (PIML) method using incomplete physics model design constraints can maintain the physics consistency of ML training results and improve vanilla ML model performance (Karniadakis et al., 2021). For this purpose, the incorporation of physics knowledge in the main parts of the ML pipeline, including the augmented input space (Q. Wang, Taal, & Fink, 2021; Chao, Kulkarni, Goebel, & Fink, 2022), the algorithm architecture (Yucesan & Viana, 2020; Viana, Nascimento, Dourado, & Yucesan, 2021) and the objective function (J. Wang, Li, Zhao, & Gao, 2020), has received

much attention in recent years in PHM.

These researches prove that we can integrate physics knowledge directly in ML, especially the Deep neural networks (Nascimento, Corbetta, Kulkarni, & Viana, 2021) and in turn, ML can compensate for incomplete physics models (Yucesan & Viana, 2022), thus establishing a mapping between structural parameters (causal factors) and degradation states (phenomena).

However, in the face of such limited data, the performance of the PIML model is yet unknown. Moreover, many PIML improvements are primarily based on over-parametric Neural Networks (Caixian, 2021) that often have more parameters than the data points available for a single training batch (Deepmind, 2019). The large number of parameters can assist in expressing the complexity of the association between the casual factor and data to some extent, but may lead to over-fitting issue and becomes infeasible for real-time applications. Meanwhile, we still lack an intuitive sense of the mechanism of physics information in ML. An interesting question will be investigated and explored in this paper is whether the vanilla non-over-parametric lightweight model has the flexibility to embed the same knowledge in different ways to achieve better performance gains.

This paper is organized as follows. Section 2 aims to present problem statement while in Section 3, we describe the simulation procedure for the stiffness deterioration of a bearing. In Section 4, three different methods for integration of physics knowledge in lightweight TCN are detailed. The performance of three proposed PI-TCN models as well as the physics knowledge’s role in training process of these models are investigated in Section 5. Finally, conclusions and perspectives of this work are discussed in Section 6.

2. PROBLEM STATEMENT

Bearing damages start from inside. Until the initial crack extends to its surface, there are no obvious signs of failure that can be observed because the geometry of the rollers is not altered. After that, the crack accelerates and the bearing fails rapidly (Khan, Kumar, Singh, & Singh, 2021). It is the root of the slight trend data. Hence, one can cite the following challenges for prediction of the bearing’s RUL based on vibration signals, illustrated by Fig.1:

1. Throughout continuous stiffness degradation of bearings, the vibration signal varies insignificantly in the early phases but changes dramatically only near the failure time. As a result, it is not trivial to capture trend information from vibration signals which reflect the ongoing degradation evolution.
2. As the stiffness degradation is hidden, when investigating historical run-to-failure data we only know the linear function of RUL in working time and do not know the duration of bearing health state and degradation state. Then, it is not trivial to match the linear RUL values with

hidden non-linear stiffness degradation process.

To address the above challenge, we propose a lightweight TCN as a benchmark purely data-driven model, using time-domain statistics as the input features to predict the bearing RUL in stiffness degradation. The input space includes mean, variance, max, min-max, root mean square, skew, kurtosis, peak factor, waveform factor, impulse factor and margin values. **The output of the TCN is a deterministic value of the RUL (days).** Then, we improve the performance of this benchmark model by integrating incomplete physics knowledge about the analytical relationship between stiffness degradation and vibration signal in terms of the augmented input space, modified hidden layer, and conflict loss function. These integration form three PITCN models: PI Features model (PIFM), PI Layers model (PILM), and PI Layer Based Loss models (PILLM).

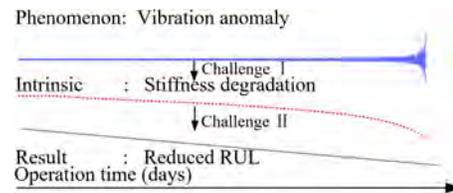


Figure 1. Challenges of bearing’s RUL prediction under stiffness degradation.

The relationship between stiffness and vibration amplitude is shown in Eq.1 (Blake, 1961), where Vib_p is the peak value of the vibration signal and $stiff$ represents the corresponding equivalent contact stiffness level. ϵ denotes the relevant imbalance in the system load. It is the extrinsic excitation of the bearing vibration. m represents the equivalent system mass. Ω is the rotation speed. In real conditions, the exact values of ϵ and m are unknown. Only the parameters Ω and Vib_p are available in vibration based RUL prediction.

$$Vib_p = \frac{\epsilon \Omega^2}{stiff - \Omega^2} \quad (1)$$

The main objective of the proposed PI-TCN models is to approximate the mapping function g between the features extracted from vibration signals and the bearing’s RUL values.

$$RUL = g\left(\frac{\Omega^2}{Vib_p}, \epsilon, m\right) \quad (2)$$

3. CASE STUDY DESCRIPTION

This case study aims to predict the RUL of a roller bearing, subject to stiffness degradation caused by the effect of crack expansion of its rollers. We assume that this bearing operates at a constant speed. And the bearing’s state is monitored via vibration signals. In subsection 3.1, we describe how to simulate continuously degraded stiffness curves while subsection 3.2 presents how to generate vibration signals.

3.1. Continuous stiffness degradation simulation

As shown in Fig.2, we generate a total of 200 bearing stiffness run to failure degradation trajectories. The mean value of the failure times of those 200 trajectories is 8.04×10^5 and its standard deviation is 1.47×10^4 . Among these trajectories, 50 sets are randomly selected as the test sets, and the remaining 150 sets are randomly divided into training and validation sets in 4:1 ratio. Each stiffness degradation trajectory is composed of health state, nonlinear degradation period and uncertain factors associated to operating environment effects.

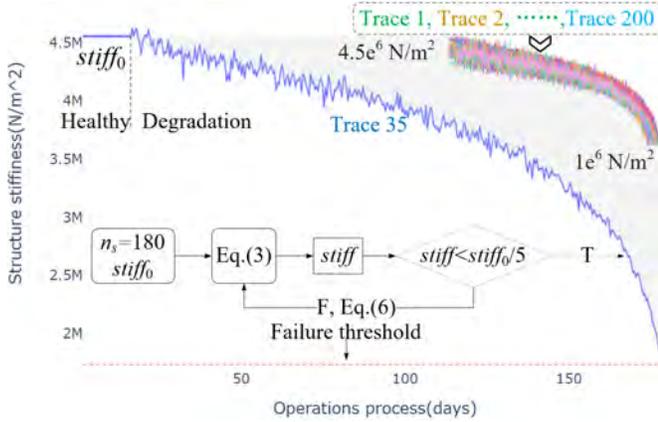


Figure 2. Simulation process of contact stiffness degradation.

We use the Eq.3 from (J. Liu & Shao, 2015) to calculate stiffness in different damage states:

$$stiff = \frac{1}{2 \left(\frac{(\cos \gamma)^{5/2}}{n_s k_p} \right)^{2/3}} + U_{stiff} \quad (3)$$

We set contact angle γ to 20° and k_p stands for the Hertz elastic contact stiffness between the ball and the smooth surface. We compute k_p according to Eq.5. The term U_{stiff} represents the stiffness model uncertainty. It is due to the complex environmental effects of the actual process, the simplifying assumptions of the model, and other factors that make the real value not strictly adhering to the physics model. We assume U_{stiff} satisfy a Skewed distribution in which the mean and the variance are equal to 10% and 5% of $stiff$ respectively. If the bearing is healthy, we calculate the stiffness via Eq.4:

$$stiff_0 = \frac{1}{2 \left(\frac{(\cos 20)^{5/2}}{180 k_p} \right)^{2/3}} \quad (4)$$

where n_s is the number of contact surfaces. Duration of health condition days generated by random seeding rand (2, 20). Bearing degradation tends to be severe when n_s decreases. In this simulation, the initial value of n_{s0} is set to 180 while the duration, in which the bearing state is healthy, is generated by a random seed as shown in Eq.4. The coefficient

k_p is computed with

$$k_p = \frac{4R^{1/2}}{6 \left(\frac{1-\nu^2}{E} \right)} \quad (5)$$

We set bearing roller radius R to 0.003 m, Poisson's ratio ν to 0.3, Young's modulus E to 2.1×10^{11} in the different simulations. We assume that the expansion of the bearing defect will result in a continuous reduction in the number of contact surfaces between the roller and raceway, with more areas changing from a face-to-face contact to a defective edge line-to-face contact, as shown in Fig.3. This process can be simulated by reducing n_s in Eq.6.

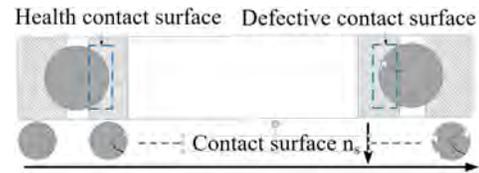


Figure 3. Defective contact and the roller failure schematic.

We generate an in-homogeneous degradation by the Eq.6. We set $steps$ to 0.0001. We use Eq.1 to get vibration amplitude, where ε is 20 g.cm, m is 5 kg. Ω takes the value 4200 rpm to indicate the wear at constant speed and external load. With U_{deg} denoting the uncertainty to reflect the non-uniform expansion of defects. U_{deg} conforms a Skewed distribution in which the mean and the variance values equal to 10% and 5% of the n_{si} respectively.

$$n_{si} = n_{s0} \times steps \times i \times (180 - steps \times i) + U_{deg}, i \in \mathbf{N} \quad (6)$$

3.2. Generation of discrete vibration monitoring signals

The stiffness values are assumed to be monitored by vibration sensors whose measurements are recorded every six hours. In this case study, the vibration amplitudes are generated by Eq.1 and then substituted into Eq.7 (if the bearing is in healthy state) or into Eq.8 (if the bearing is in degradation state) to obtain the vibration sequences. Shocks caused by defects are commonly accompanied by both frequency and amplitude noise. We add frequency noise (n_{d1}) according to the Skewed distribution of (3, 1). With 3 as the mean and 1 as the variance, this indicates an uncertainty shock due to a roller defect introducing a high multiple of the rotational frequency during the bearing rotation. We also add amplitude noise (n_h) according to the signal-to-noise ratio of 1/100 in the healthy state and 1/10 in (n_{d2}).

$$x(t) = vib_p \times \sin\left(\frac{2\pi\Omega}{60}t\right) + n_h \quad (7)$$

$$x(t) = vib_p \times \sin\left(\frac{2\pi\Omega}{60}t\right) + vib_p \times \sin\left(\frac{2\pi\Omega}{60}t(1+n_{d1})\right) + n_{d2} \quad (8)$$

Each vibration sample is generated according to 4096 Hz sampling frequency of 3 s time length. More than 700 vi-

bration samples were collected for each trajectory, as shown in Fig.4. It can be seen that the generative data satisfy the non-trending characteristics.

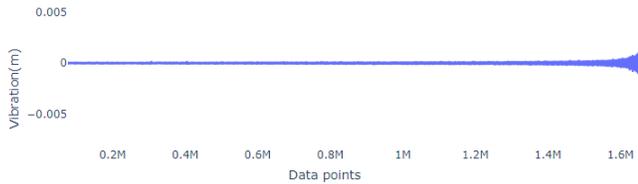


Figure 4. Illustration of the generated vibration signal.

4. DESCRIPTION OF THE PROPOSED METHODS

Informed learning is the seamless incorporation of constraints in an ML pipeline. PIML means transforming physics knowledge as potential or direct constraints of ML. In PHM, knowledge is parametrically representable perceptions of system behavior and failure mechanisms. In this section, we will present the purely data driven-model, i.e., a lightweight TCN model to predict bearing RUL, and then propose three PITCN models based on the same knowledge to improve the prediction performance. The first one is the PI Feature model (PIFM), which guides the extraction of data features through parameter relationships in the physics formulation. The second one is the PI Layer model (PILM), which embeds the physics input-output model in the computational function of the layer. And finally, the PI Layer Based Loss model (PILLM) adds regularization terms associated with the output of the physics model to the loss function of the vanilla TCN.

4.1. Purely data-driven model

CNN allows parallel computation of outputs and thus can achieve better performance than RNNs in sequence modeling (Lea, Flynn, Vidal, Reiter, & Hager, 2017). TCN, consisting of dilated, 1D convolution layers with the same input and output lengths, avoid common pitfalls of recursive models, such as gradient explosion or disappearance problems or lack of retention. We build lightweight TCN as “Benchmark” based with causal separable Conv1D, in Fig.5. The total number of parameters in the model is 3,411. Note that the subsequent implementation of PI-TCN will compute the physics features related to stiffness in the hidden layer, the values of which have a high probability of putting the neurons in the saturation zone. Therefore, the activation function is chosen for the nonlinear function h-swish with no upper bound, lower bound, smooth, and non-monotonic characteristics.

4.2. Physics-informed feature augmented input space

Eq.1 inspires us that the factor Ω^2/Vib_p can be used as a physics feature to predict the RUL of the bearing due to the RUL’s dependence on the stiffness degradation level. Among the original 11 time-domain statistical features, we removed the Max feature while physics-informed health indicator (PHI)

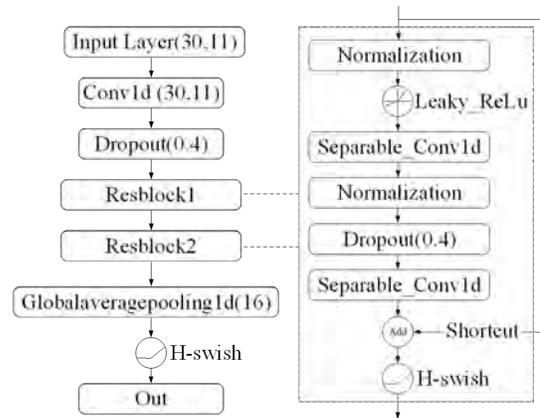


Figure 5. Lightweight TCN architecture diagram.

is added to construct new input samples having the same dimension (60×11) as the benchmark model. The benchmark model shown in Fig.5 is then re-trained as shown in Fig.6.

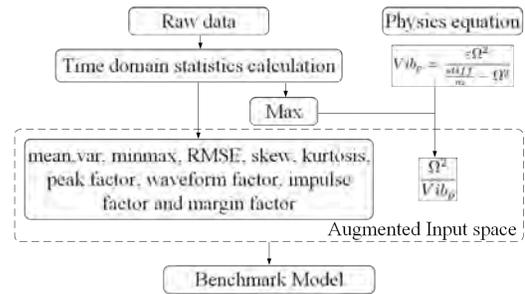


Figure 6. Create PIFM based on physics analytic relationships.

Although Ω^2/Vib_p is not an exact stiffness estimation, it has a clear physical meaning for being self-compiled quantities as stiffness. It contains trend information. PIFL is essentially an extension of the series combination structure to create a hybrid model. The output based on the physics model is part of the TCN input. The PI layer adds potentially physically consistent weak constraints to the input space through physics model parameter relationships based feature extraction.

4.3. Physics embedded layer

Fault dynamics models can be converted into an input-output module in ML, and in turn ML compensates for model incompleteness, as shown in Fig.7.

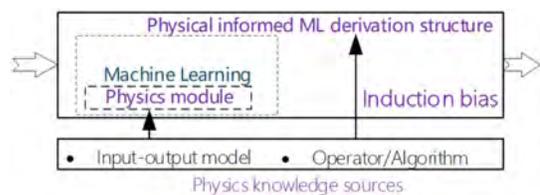


Figure 7. Embedding physics equations in NN layer.

Based on this paradigm, we have developed the PILM. Particularly, Eq.1 is transformed into a neural network linear model in Fig.8. The unknown function $g(\cdot)$ in Eq.2 can be approximated using a custom layer function $h(\cdot)$ of the neural network in the structure presented by Fig.9.

This model allows extracting the PHI Ω^2/Vib_p by embedding the transformation layer of Eq.1. Then, the PHI is used as the input of the hidden custom layer whose structure is defined as an approximate function of stiffness degradation in Fig.8. In the training process, the unknown parameters ε , m and U_{stiff} reflected by the weights (ω) and biases (b) of the hidden layer are updated to optimize the prediction results.

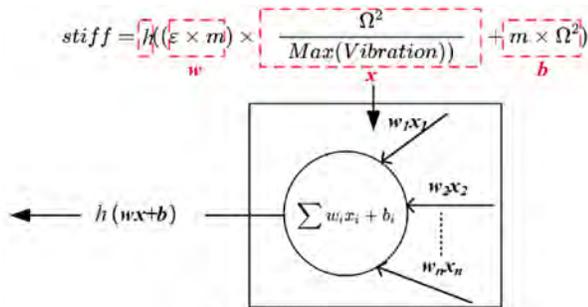


Figure 8. Embedding physics equations in NN layer.

Compared to the PIFM, PILM provides induction bias for TCN, as shown in Fig.7. It is able to ensure that the computational process based on physics knowledge is forced during the data processing of TCN, thus completely embeds the physics knowledge into the computational paradigm and overall derivation process of the TCN model.

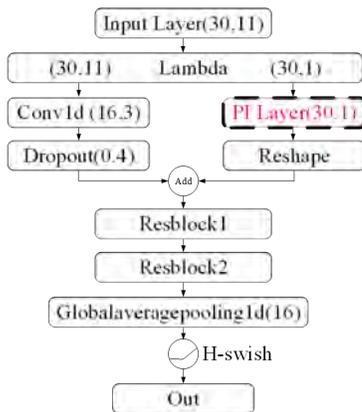


Figure 9. PI Layer outputs join the Resblocks' training.

4.4. Physics-informed layer based conflict loss

We also introduce the physics inconsistency by designing the loss function according to conflict between physics model's output and ML output. The whole PILLM consists of two parts: branch network and main network. An output layer is added after the physics informed layer to provide the PHI in the branch network. These features participate in the training process of the main network at one hand, and influence the

hyper-parameter optimization of the main network through the loss function on the other hand, as shown in Fig.10. The outputs of each are measured with two losses and assigned corresponding weights of 1.0 and 0.2. In the prediction process, only the prediction results of the main network are utilized.

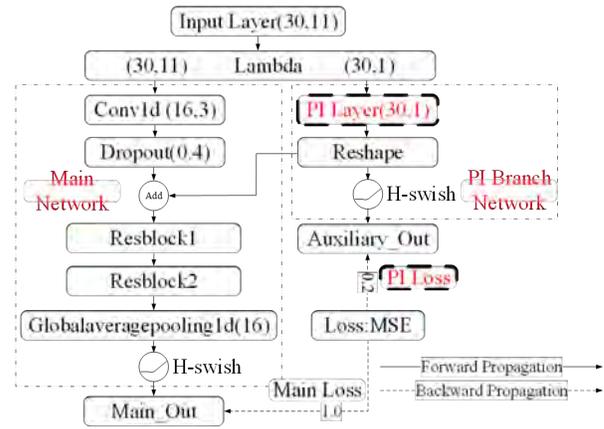


Figure 10. Build PI-loss based on different branches conflict.

In contrast to the first two methods, PILLM aims to intervene in the adaptive search process of the TCN in the solution space. Firstly, due to the single finite feature, it can be conjectured that the prediction accuracy of the PI Branch network is limited, resulting in high level loss, so the feature of minimizing the loss function in the training process of the TCN is used to improve the prediction accuracy of the Main network part. Secondly, by allowing the PHI of the PI branch network to participate in the training process of TCN, the two parts are permitted to share optimization process, thus allowing the TCN to satisfy a certain degree of physics consistency.

5. RESULT DISCUSSION

We fix the training epochs to 1000, design an early stop mechanism with patience equal to 80 epochs. We initialize the network parameters with uniformly drawn weights. The batch size of 128. The input-shape of each batch is (30,11). All models are trained in the same conditions. Moreover, we use Adam Optimizer in training (Kingma & Ba, 2014).

5.1. Investigation of the PITCN models' performance

Fig.11 presents different model's prediction results through 10 randomly selected trajectories of the test set while Fig.12 shows the box plots of the differences between the predicted and the truth RUL on the overall test set. The results highlight the performance of the PIML models compare to the one of the purely data-driven model: the same physics knowledge has different incorporation possibilities and potential for improving the benchmark. Particularly, we find that the PILLM model has the best performance. Its error range is only [17.97, 15.65] while the ones of the BENCHMARK, PIFM and PILLM

are respectively [-95.51, 79.83], [-26.85, 36.66], and [-33.38, 26.11].

Table 1 presents the performance of the proposed models compared with the benchmark on the overall test sets. We get the following conclusions from Fig.11, and Fig.12:

1. PI-TCN models show more accurate prediction with the smaller prediction error limits compared to Benchmark.
2. The predicted results of PI-TCN converge with the trend in the real value, showing the possibility to effectively solve the “Challenge II” in Fig.1.
3. Among the three different PI-TCN models, PILLM has the best prediction stability with the most compact upper and lower error limits and the minimum error mean, as shown in Fig.12.

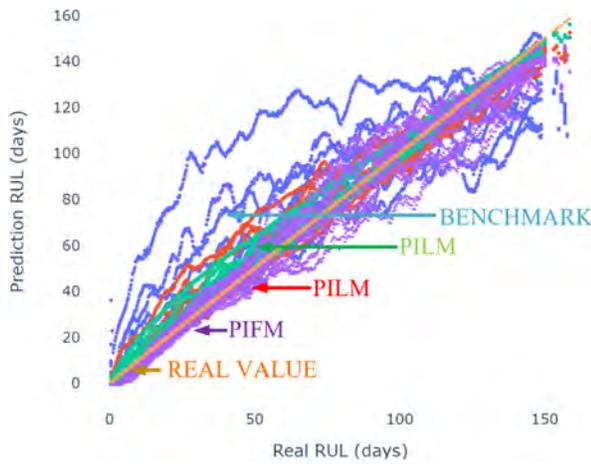


Figure 11. Prediction results of different models.

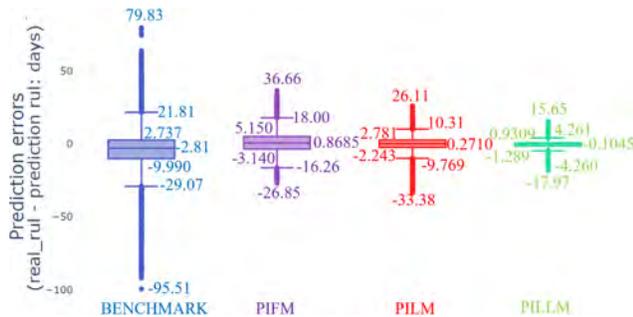


Figure 12. Model evaluation results.

Table 1. Performance evaluation on the test dataset.

	Benchmark	PIFM	PILM	PILLM
RMSE	16.191	8.059	5.818	3.157
MAE	10.878	5.939	3.942	1.965
R2	0.862	0.959	0.982	0.994

5.2. Investigation of physics knowledge’s role in ML

To investigate the role of physics knowledge in the training process of the proposed models, we build the channel information model to generate the channel heat-map. After training, we extract the layers in which physics knowledge is integrated to investigate the correlation between physics information and results. More concretely, for the PIFM, we choose the input layer as the channel information model while for the PILM and PILLM, we choose the input layer as input and the “Add” layer as output. The brightness of the colors in the heat map reflects the correlation. The obtained results are presented in Fig.13, 15, and 16.

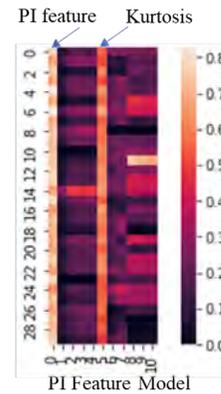


Figure 13. Weight heat-map of the input layer in PIFM.

In the PI Feature model’s channel information heat-map, we find that the model focuses more on the channels where the Ω^2/Vib_p and Kurtosis are located. Kurtosis as a higher order statistic with the ability to capture dramatic trends from flat data. However, the Ω^2/Vib_p feature are assigned a higher weight than Kurtosis. This result highlight the intuition that the physics-informed feature generated based on analytic relational formulations provide additional crucial information to improve the model performance.

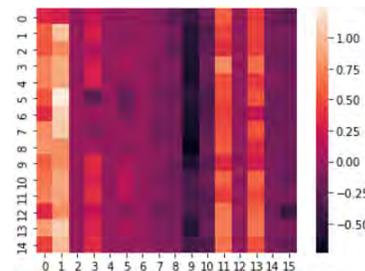


Figure 14. Heat-map of Benchmark model.

The results of Benchmark Conv1d layer are selected to build the channel information model, and the heat map is generated as shown in Fig.14, which is used as the cross-sectional com-

parison object of PILM and PILLM.

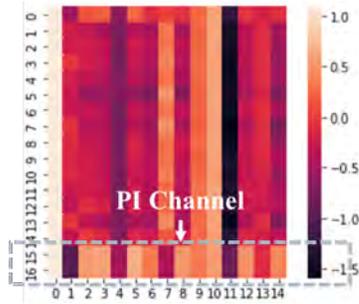


Figure 15. Heat-map of the PILM “Add” Layer.

For the heat-map of PILM, it can be seen in Fig.16, the features flowing from the PI Layer receive higher attention relative to the other channels.

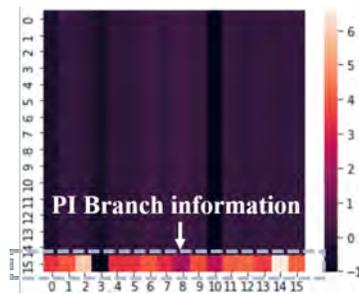


Figure 16. Heat-map of the PILLM “Add” Layer.

For the PILLM’s heat-map, as shown in Fig.16 the part incorporated into the feature from the PI Loss is dominant compared to the other channels. In contrast to the feature map of the benchmark on the same layer, the feature map of the model is dimensionless, with a size of 16 x 15, because the features of PI Loss are not incorporated. The overall value of this feature map is smaller than the corresponding value of PI Loss. So this result highlights that physical knowledge plays a significant role in improvement of the prediction results.

6. CONCLUSIONS

In this paper, the physics knowledge about the relationship between vibration signals and stiffness degradation is exploited to create the physics-informed TCN models in three ways: augmented input space, physics equation embedded layer, and physics-informed conflict loss. The simulation result demonstrates the flexibility of methods incorporating physics knowledge and also highlight the significant improvements they can bring to vanilla TCN when working with “slight trend” data. In comparison with the benchmark model, PIFM, PILM, and PILLM reduce the mean prediction error by 69.39% (from 2.81 days to 0.8685 days), 90.35% (from 2.81 days to 0.2710 days) and 96.29% (from 2.81 days to 0.1045 days), respec-

tively. By investigating channel weights in the related layer, we found that those improvements mainly stem from the focus of three PIML models placed on data streams containing physical information. Indeed, the underlying logic of the PIML models is to guide ML to capture features related to degradation by encoding physics knowledge and then to establish the underlying relationships between features and RULs thanks of ML’s non-linear mapping capabilities. In our case, the stronger the physics constraints imposed by the encoding in TCN, the better the PITCN model performs. In future works, we will use sparse noise monitoring data and conduct in-depth research on the translatability of incomplete physics knowledge containing uncertainty to ML pipeline. Furthermore, PIML methods will be developed in real scenarios with complex systems and complex operating conditions.

REFERENCES

- Blake, R. E. (1961). Basic vibration theory. *Shock and vibration handbook*, 1, 2–8.
- Caixian, C. (2021). Halving the error rate requires more than 500 times the computational power. <https://chowdera.com/2021/10/20211024171246242z.html>.
- Chao, M. A., Kulkarni, C., Goebel, K., & Fink, O. (2022). Fusing physics-based and deep learning models for prognostics. *Reliability Engineering & System Safety*, 217, 107961.
- Deepmind. (2019). Alphastar: Mastering the real-time strategy game starcraft ii. <https://www.deepmind.com/blog/alphastar-mastering-the-real-time-strategy-game-starcraft-ii>.
- Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6), 422–440.
- Khan, S., Kumar, R., Singh, M., & Singh, J. (2021). Vibration and acoustic method for detection of cracks in bearings: A critical review. *Advances in Engineering Design*, 221–229.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Lea, C., Flynn, M. D., Vidal, R., Reiter, A., & Hager, G. D. (2017). Temporal convolutional networks for action segmentation and detection. In *proceedings of the ieee conference on computer vision and pattern recognition* (pp. 156–165).
- Li, F., Hu, W., Meng, Q., Zhan, Z., & Shen, F. (2018). A new damage-mechanics-based model for rolling contact fatigue analysis of cylindrical roller bearing. *Tribology International*, 120, 105–114.
- Liao, L., Jin, W., & Pavel, R. (2016). Enhanced restricted boltzmann machine with prognosability regularization for prognostics and health assessment. *IEEE Trans-*

actions on Industrial Electronics, 63(11), 7076-7083.
doi: 10.1109/TIE.2016.2586442

- Liu, H., Song, W., Zhang, Y., & Kudreyko, A. (2021). Generalized cauchy degradation model with long-range dependence and maximum lyapunov exponent for remaining useful life. *IEEE Transactions on Instrumentation and Measurement*, 70, 1–12.
- Liu, J., & Shao, Y. (2015). A new dynamic model for vibration analysis of a ball bearing due to a localized surface defect considering edge topographies. *Nonlinear Dynamics*, 79(2), 1329–1351.
- Massi, F., Bouscharain, N., Milana, S., Le Jeune, G., Maheo, Y., & Berthier, Y. (2014). Degradation of high loaded oscillating bearings: Numerical analysis and comparison with experimental observations. *Wear*, 317(1-2), 141–152.
- Nascimento, R. G., Corbetta, M., Kulkarni, C. S., & Viana, F. A. (2021). Hybrid physics-informed neural networks for lithium-ion battery modeling and prognosis. *Journal of Power Sources*, 513, 230526.
- Porotsky, S., & Bluvband, Z. (2012). Remaining useful life estimation for systems with non-trendability behaviour. In *2012 IEEE conference on prognostics and health management* (pp. 1–6).
- Shi, Z., & Chehade, A. (2021). A dual-lstm framework combining change point detection and remaining useful life prediction. *Reliability Engineering & System Safety*, 205, 107257.
- Viana, F. A., Nascimento, R. G., Dourado, A., & Yucesan, Y. A. (2021). Estimating model inadequacy in ordinary differential equations with physics-informed neural networks. *Computers & Structures*, 245, 106458.
- Wang, J., Li, Y., Zhao, R., & Gao, R. X. (2020). Physics guided neural network for machining tool wear prediction. *Journal of Manufacturing Systems*, 57, 298–310.
- Wang, Q., Taal, C., & Fink, O. (2021). Integrating expert knowledge with domain adaptation for unsupervised fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*.
- Yucesan, Y. A., & Viana, F. A. (2020). A physics-informed neural network for wind turbine main bearing fatigue. *International Journal of Prognostics and Health Management*, 11(1).
- Yucesan, Y. A., & Viana, F. A. (2022). A hybrid physics-informed neural network for main bearing fatigue prognosis under grease quality variation. *Mechanical Systems and Signal Processing*, 171, 108875.

BIOGRAPHIES



Deng WeiKun received the M.S and Bachelors degree in aerospace propulsion theory and engineering from Northwestern Polytechnical University University, Xi'an, China, in 2020. He a Ph.D researcher at Ecole nationale

d'ingénieurs de Tarbes-Toulouse INP, France. His current research interests include machine learning based prognostics and health management and rotor dynamics.



Khanh T. P. Nguyen is Associate Professor at National School of Engineering in Tarbes (ENIT), France, and member of the Production Engineering Laboratory (LGP) since 2017. She received the Ph.D. degree in Automation and Production Engineering from Ecole Centrale de Nantes, France in 2012. From 2013 to 2015, she was a Postdoctoral

Fellow with the French Institute of Science and Technology for Transport, Development and Networks (IFSTTAR). From 2016 to 2017, she was an Assistant Professor with the University of Technology of Troyes. Her research interests include applications of artificial intelligence in predictive maintenance and prognostics and health management (PHM).



Christian Gogu received his PhD in 2009 as part of a joint PhD program between the Ecole des Mines de Saint Etienne (France) and the University of Florida. He is currently Associate Professor in the department of Mechanical Engineering at Université de Toulouse and does his research within the Institut Clément Ader (ICA). His research interests include design under uncertainty, multidisciplinary design optimization, machine learning based diagnostics and prognostics with applications mainly to aerospace structures.

design under uncertainty, multidisciplinary design optimization, machine learning based diagnostics and prognostics with applications mainly to aerospace structures.



Jérôme Morio is research director at ONERA - the French Aerospace Lab - in Toulouse (France). He received his Ph.D in image processing from Aix-Marseille University (France) in 2007. His main research interests include uncertainty management, rare event probability estimation and sensitivity analysis.



Kamal Medjaher received the Ph.D. degree in control and industrial computing from University of Lille 1, Villeneuve-d'Ascq, France, in 2005. He was Associate Professor at the National Institute of Mechanics and Microtechnologies, Besançon, France, and FEMTO-ST Institute, from 2006 to 2016. He is currently Full Professor at Tarbes National School of Engineering (ENIT), France. He conducts his research activities within the Production Engineering Laboratory. His current research interests include prognostics and health management of industrial systems and predictive maintenance.

He is currently Full Professor at Tarbes National School of Engineering (ENIT), France. He conducts his research activities within the Production Engineering Laboratory. His current research interests include prognostics and health management of industrial systems and predictive maintenance.

Design and validation of scalable PHM solutions for aerospace on-board systems

Fabio Federici¹, Cecilia Tonelli¹, Mathieu Le Cam², Marcello Torchio², and David Larsen³

Collins Aerospace, ¹ Rome, Italy, ² Cork, Ireland, ³ Vergennes, VT, United States of America

{name.surname}@collins.com

ABSTRACT

In recent years, Prognostic & Health Management (PHM) has become a topic of strong interest in the aerospace domain. Health assessment and remaining useful life estimation for on-board systems provide several advantages, mainly related to the increased analysis capabilities and the reduction of maintenance interventions (and, consequently, of operating costs). For this reason, it is of interest for the aerospace industry to identify and define efficient strategies both for the introduction of native PHM capabilities in new generation on-board systems and for the retrofit of existing ones. This paper proposes a strategy for the scalable deployment of PHM techniques for on-board systems, with particular focus on edge computing capabilities. Different reference scenarios (ranging from cloud-based processing to local-only processing) are presented, and an edge-focused PHM architecture is discussed in detail, with the relative challenges addressed. The design and validation of proposed edge-based solution is described, with specific reference to its support for an existing data analytics framework. The solution is then assessed against a reference aerospace use case involving a representative aircraft braking system, focusing on computational aspects to highlight the compatibility of the proposed deployment strategy with efficient on-board computations.

1. INTRODUCTION

Prognostics and condition-based maintenance (CBM) have attracted significant interest of the aerospace sector in the recent years. The goal of prognosis is to track degrading aspects of the overall design to predict deviation with respect to a reference baseline (e.g., healthy condition). Generally, we define prognostic systems as ones that compute remaining useful life (RUL), performance life remaining (PLR) or state of health (SOH) with sufficient fidelity and sufficient advance notice to allow a maintenance action well before an

operational failure (SAE JA6268). Prognostics may provide significant benefits when applied to complex aerospace systems, potentially reducing maintenance-related downtime and costs, and improving the overall efficiency of the systems.

Traditional approaches to prognostics relied on physics-based methods (Cadini, Zio, & Avram, 2009) often involving fault propagation and reliability models for the component or system under consideration. Such methods require a thorough understanding of the system, and they are usually specific to a component and not generalizable for a broad variety of applications. More recently, data-driven approaches emerged as a powerful alternative. These approaches perform RUL prediction from the operational run-to-failure raw time series data, collected from sensors mounted on the components or systems under consideration. There are two types of data-driven approaches in the literature, *direct* and *indirect*.

Direct approaches rely on training a neural network to learn the RUL directly from the run-to-failure time series data (Zhang, Wang, Li, Cui, Liu, Yang, & Hu, 2018).

Indirect approaches rely instead on the so-called health monitoring, i.e., mapping the time series data into a one-dimensional Health Index (HI), which decreases monotonically and proportionally to the time series degradation (Mosallam, Medjaher, & Zerhouni, 2015). Deep learning methods are used frequently (Reddy, Venugopalan, & Giering, 2016) for health monitoring purposes. Once the health index is computed, RUL can be estimated, for instance, as a weighted average of RULs of matching HI curves (Wang, 2010) of all the time series in the training dataset.

One of the more relevant problems related to the computation of RUL for aerospace systems is the reduced availability of operational data from the systems under consideration. An efficient prognostics framework should be able to acquire data from sensors during relevant periods, collecting extended time series and provide them as input to described algorithms, which should be in turn deployed over proper computation platforms. Those capabilities are generally framed in the wider Integrated Vehicle Health Management

Fabio Federici et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

(IVHM) capability, which could be intended as the supporting platform enabling CBM of complex aircrafts. Airborne systems include a variety of hardware and software components, complicating the definition of standardized mechanisms and infrastructures supporting CBM. Over the years, several standards (e.g., ISO 13374 or SAE JA6268) and architectural specifications (e.g., Open System Architecture OSA-CBM) tried to provide guidance for the implementation of generic, interoperable architectures (Chang, Gao, & Wang, 2018) (Goebel & Rajamani, 2021). Several authors presented implementations compliant, or inspired to cited reference standards, providing mappings over specific supporting technologies. Tambe (2013) presented a distributed architecture for avionics sensor health assessment compliant with OSA-CBM and using the Data Distribution Service (DDS), a standard for real-time distributed systems supporting the publish/subscribe paradigm. Ezhilarasu and Jennions (2021) developed an architecture of a Framework for Aerospace Vehicle Reasoning (FAVER), a system-agnostic framework inspired by OSA-CBM and developed to isolate propagating faults by incorporating Digital Twins (DTs) and reasoning techniques. In other cases, the reference standards have not been considered, opting for alternative architectural proposals. Wang, Pan, Xiong, Fang, and Wang (2017) presented a software architecture based on Service-Oriented Architecture (SOA) and dual bus technology to share the information from on-board systems. Chen, Hu, & Hou (2021) proposed a general, time-variant architecture model usable to simulate PHM systems. Li, Verhagen, & Curran (2020) discussed a generic architecture along with a systematic methodology to support the design of PHM systems.

Only a limited number of works considered the integration of high-performance computing unit on-board the aircraft. For example, Chen, Liu, & Zhou (2020) presented a VPX-based computing platform for aircrafts with an artificial intelligence module built-in.

From a design perspective, several choices must be addressed to define a system supporting outlined functionalities (Li et al., 2020).

Differently from the literature presented above, this paper proposes a strategy for the scalable deployment of PHM techniques for on-board systems, with particular focus on edge computing capabilities. Different reference scenarios (ranging from cloud-based processing to local-only processing) are presented, and an edge-focused architecture, along with the challenges related to its implementation, is discussed. The selected solution is then detailed and evaluated with specific reference to the possibility of supporting an existing, mature data analytics framework. This integrated solution is then assessed against a reference aerospace use case involving a representative aircraft braking system, detailing its initial validation, and analyzing the feasibility of proposed system for a real-world deployment.

2. PHM SUPPORT FOR ON-BOARD SYSTEMS

A CBM support system usually includes multiple functions. As a reference, the OSA-CBM functional model (MIMOSA, 2022) is represented in Figure 1.

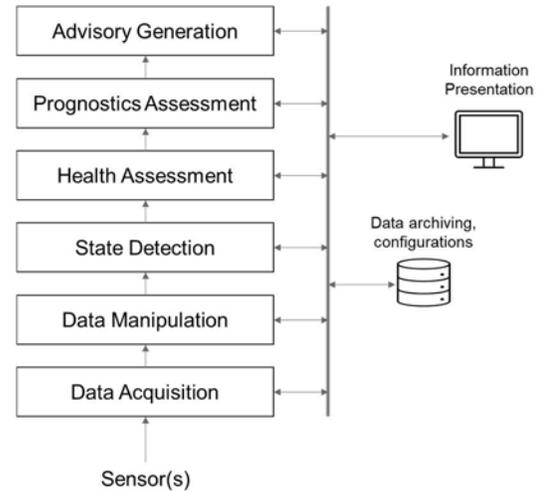


Figure 1 - OSA-CBM reference functions

The basic functionalities required by a CBM support system are two: (i) Data Acquisition (DA), and (ii) Data Manipulation (DM). The DA functionality provides access to the raw sensors' data collected from a Target System (TS) and may additionally include simple sensing calibration capabilities. The DM functionality, instead, implements mainly signal processing functions, together with sensor fusion and feature extraction algorithms. The State Detection (SD) function is used to estimate the current condition of the TS. It usually includes various functions ranging from Built-in Test (BIT) to more advanced components oriented to fault-detection. The Health Assessment (HA) function provides a SOH estimate of the TS, usually considering its operating conditions as well as the results of previous assessments. The Prognostics Assessment (PA) function estimates the RUL of the TS, either directly or indirectly (i.e., relying on the SOH estimate provided by the HA function).

This work mainly focuses on DA and DM, along with the integration of the HA and prognostic functions. The Advisory Generation (AG) function, that usually provides reports and recommendations for maintenance actions based on the results of the HA and PA, is not considered in this work.

The described functions shall be deployed onto suitable computing platforms and integrated with the existing aircraft systems, such as the TS (to ensure DA from sensors), but also other systems (or sub systems) that enable data transmission, aggregation, and processing. Ideally, the integration of the highlighted functions should have minimal impact on the overall aircraft design, especially in terms of Size, Weight, And Power (SWAP) consumption. The overall CBM support system should also be designed to support modularity and

versatility in terms of integration with other systems (Finda & Hédl, 2014). CBM functions are frequently delivered through a distributed architecture, that can potentially involve both on-board and ground-based operations. Figure 2 represents a high-level view of a possible deployment for CBM related functions.

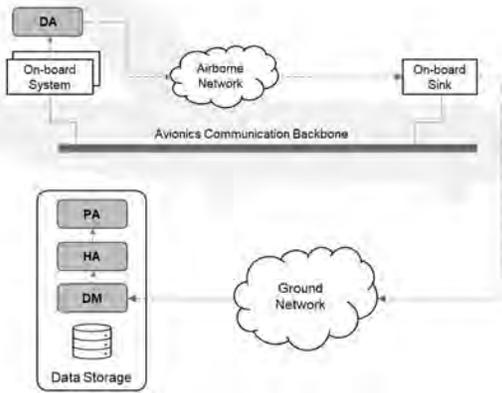


Figure 2 – Schematization of a ground-based CBM supporting architecture

In particular, the setup in Figure 2 represents a generic architecture that enables ground-based CBM operations through a mix of on-board and ground-based systems. The on-board system is only responsible for providing DA functions that collect measurement from the TSs. The TSs' data are acquired and collected during flight, continuously or only during relevant flight phases. The collected data are transmitted towards an on-board aggregation unit (or sink) through an airborne network. It should be noted that this network is generally a dedicated communication infrastructure, distinct from the traditional avionics' communication backbones. Finally, the collected data are transferred to a ground-based infrastructure through a ground network. Data may be streamed continuously (e.g., satellite connections), or stored for the flight duration and transferred once the aircraft has landed. The ground-based infrastructure is responsible to carry out all the remaining functions of the OSA-CBM reference model in an offline fashion (i.e., without processing the data as they are received). The data may be permanently stored and analysed according to specific HA and prognostics techniques. Modern deployments may leverage private cloud infrastructure to support the ground-based operations (Terrissa, Meraghni, Bouzidi, & Zerhouni, 2016).

This approach presents several advantages in terms of implementation: it requires a limited number of functions to be deployed onto on-board systems, and the ground-based offline computation is not constrained by limited resources in terms of computational power and storage. On the other hand, this architecture requires the collection of extended time-series data, which must be transmitted and temporarily stored

on-board (e.g., by the aggregation unit), and then transmitted to the ground infrastructure. Moreover, the availability of ground-based facilities to support the required computations is generally not guaranteed for every type of aircraft.

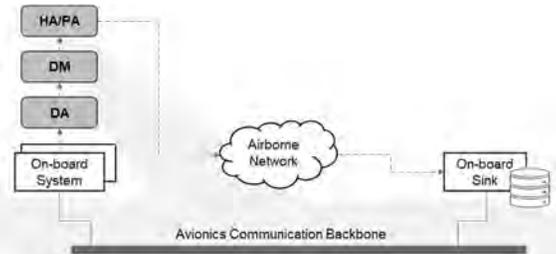


Figure 3 - Schematization of an edge-oriented CBM supporting architecture

A possible way to overcome the aforementioned limitations could be through the adoption of edge-oriented architectures. Figure 3 provides a representation of this scenario. This setup supports the on-board execution of target CBM strategies. Indeed, the on-board system would not only support DA functionalities, but would also provide extended computing capabilities to host other, more advanced functions, as DM, HA, and PA. Data are acquired and processed during flight, continuously or only during relevant flight phases, in order to produce relevant results related to RUL or HA. Only the obtained results are transmitted toward the on-board aggregation unit. This setup allows to overcome several disadvantages of a ground-oriented infrastructure. The majority of CBM related functions could potentially be hosted onto the on-board system, without the need for ground-based support. The acquired data can be processed close to the source, without requiring the storage or transmission of large-size data sets, neither to an on-board sink nor to a ground-based station. As a drawback, this approach requires extended capabilities (e.g., increased computational power) to be included among on-board systems.

Intermediate scenarios could be also considered. As an example, the edge processing capability could be limited to DA and DM, with DM delivering only feature extraction functionalities. This configuration has the advantage of requiring reduced processing resources, while still allowing to support significant data reduction, as time series are processed to extract only relevant indicators.

The introduction of CBM support at the edge, from basic DA functionalities to full-fledged support for data analytics, poses multiple challenges in terms of overall system design. The following section analyses the most relevant challenges and introduces a reference architecture suitable to support the reviewed scenarios.

2.1. PHM for on-board systems: proposed approach and related challenges

All the architectures reviewed in the previous section, require the baseline capability of extracting data from the TS. This may represent a significant challenge in aerospace systems, both in the case of legacy equipment and in the case of newly designed platforms. Legacy equipment is usually not designed to support expansion for adding the additional functionalities required to support data collection, nor it provides the communication interfaces required to share collected data with other on-board systems. New equipment can be designed considering native support for CBM related functions, like DA, storage, and communication, with limited impact on overall size, weight, and power. Adding CBM-related function to safety-critical equipment (for example, including health-monitoring related items in an Engine Control Unit) can also be challenging from the point of view of safety-related certification, as these functions are typically non-critical (e.g., classified at DAL-E according to RTCA DO-178C). A system including both safety-relevant functionalities and CBM related functionalities on the same platform would be considered a mixed-criticality system and should provide guarantees of strong isolation between the different DAL levels to be certifiable.

An intermediate solution is the addition of external dedicated unit, specifically designed to support CBM related functions. This option would be potentially able to fit both the use case of legacy equipment, and that of newly developed systems, with limited impact on certification issues. The CBM functionalities would be in fact allocated to a separate platform, with proper interlock mechanisms in place to guarantee the absence of undesired interference at the communication interfaces. The platform could not only implement DA (passively receiving data transmitted by the TS over a dedicated interface, if this is supported by the design, or actively interrogating the TS to extract relevant data using available access mechanisms, such as dedicated test interfaces), but also support local data processing (from DM for feature extraction, to local HA). The obvious drawback of an external unit is the impact in terms of additional SWAP, which could be limited at the expense of reduced computational resources. Interestingly, an external dedicated unit can deliver more flexibility from the point of view of on-board communication, providing support for multiple interfaces, and allowing to run dedicated communication stacks and middleware.

This work considers the use of a dedicated unit to support edge-based CBM-related functions. Figure 4 provides an overview of the reference architecture used in the context of this work, with specific focus on the interaction among the different systems involved.

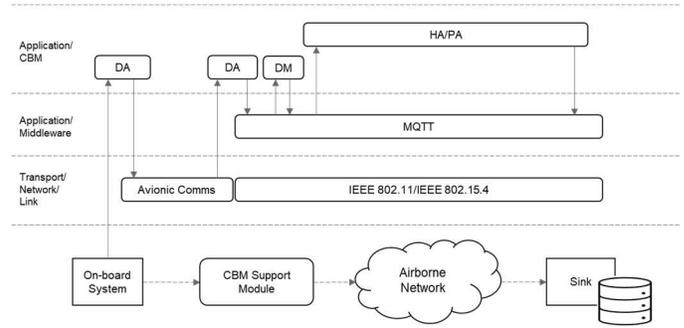


Figure 4 - High-level view of proposed system architecture

The proposed schematization identifies several reference layers, mainly related to the actual technologies supporting the distributed deployment. The lowest level includes relevant physical systems installed on-board, and related communication means. On top of this level, networking technologies and protocol stacks are considered. The application level is then split into two different layers: the middleware layer, including all components responsible for communication abstraction and the CBM layer, including the different CBM functions detailed in section 2.

As anticipated, a dedicated unit, the CBM Support Module (CSM), is connected to the TS, and it can collect relevant data from it.

The CSM communicates with the TS using a dedicated avionic interface. Data exchange is usually managed by means of a standard avionic protocol (e.g., ARINC 429), or by means of a custom protocol in case of proprietary interfaces.

The CSM also includes an external network interface, used to access the dedicated CBM support network. Different approaches can support the communication, ranging from traditional wired networks to wireless-based solutions. To reduce the overall impact of the CSM, the use of wireless technologies (e.g., IEEE 802.11, IEEE 802.15.4) may be considered.

The external network is used to communicate to a network sink module, intended as an on-board data collection unit.

The Message Queuing Telemetry Transport (MQTT) middleware provides an abstraction over the low-level communication method and allows to implement a publish-subscribe message transport between multiple on-board units. The MQTT model requires a message broker, i.e., a server able to receive messages from a sender and route it to the intended receiver. Multiple clients can connect to the broker over a network to exchange messages. In the presented approach, the CSM can run one or more MQTT clients, each one implementing a specific function. In a basic setup, only one message broker is included, running on the (e.g.) network sink module. In more advanced configurations, the CSM may host multiple functions, communicating by means of the MQTT middleware. In that case, a local broker runs on the

module, leveraging the possibility offering by MQTT of bridging multiple brokers in a network.

In the MQTT model, a hierarchy of topics support the exchange of data along the network. Clients publish new data over a certain topic and may subscribe to other topics to receive data. The broker oversees managing data distribution among the client that have subscribed a certain topic. The proposed model does not rely on statically configured clients, assuming instead that each client requires a configuration at startup. The configuration specifies which data shall be acquired by the client, the origin of this data and the means supporting data acquisition. The configuration also specifies the destination of data produced by the client (e.g., the topic over which the data shall be published). The use of configurations allows to implement re-usable modules, with a fixed functionality potentially configurable to accommodate the needs of different application scenarios.

Another advantage of the abstraction provided by the MQTT middleware is the possibility of relocating functions on different nodes of the network. For example, in the approach under analysis the DM function and the prognostic function are intended to receive and send data using the MQTT publish-subscribe mechanism. This means that those functionalities can be easily moved between the CSM and the sink module. The mapping is usually done according to specific non-functional requirements, as discussed in the beginning of this section.

Functions with specific dependencies on the underlying physical equipment have less flexibility. For example, on the CSM the data acquisition function only relies on MQTT mechanisms to share acquired data, however it requires physical connection to the TS.

3. A REFERENCE USE CASE: THE TRAJECNETS CBM FRAMEWORK

The general strategies described in section 2 can be used to map an actual CBM framework to a proposed high-level architecture and drive the detailed design of some of its reference modules. This section provides an overview of the reference framework adopted in the context of this work from the point of view of data analytics. The different CBM - related functions will then be identified and mapped onto the reference architecture in Figure 4, also detailing their specific implementation in hardware and software.

3.1. Overview of target Analytics

The proposed CBM framework is built on top of the TrajecNets approach (Shahid & Ghosh, 2019), and leverages a Recurrent Neural Network (RNN) based autoencoder for embedding the run-to-failure time series sensor data in a 2D feature space. The embedding is in the form of a trajectory representing the temporal evolution of data from healthy to failure states. This trajectory can be used for health monitoring, which can in turn be used for RUL estimation.

Figure 5 provides a high-level schematization of the data analytics pipeline for the proposed RUL estimation.



Figure 5 - Schematization of a reference data analytics pipeline

Input variables are acquired and may be pre-processed in different ways depending on the specific flight phase. The selection of the relevant flight phases and the related pre-processing strategies usually depend on domain knowledge and preliminary data exploration.

For each pre-processed input variable, up to 13 features may be computed: mean, standard deviation, variance, minimum, maximum, median, sum, mean absolute deviation, skewness, kurtosis, number of null values, 10th percentile, and 90th percentile. These features are extracted to statistically describe the variables' range of variation and helps in providing an efficient representation of the TS by reducing the amount of raw data.

The data-driven approach developed for the RUL estimation of a TS is based on an Auto-Encoder (AE) architecture such as the one presented in Figure 6. Starting from high-dimensional inputs set (i.e., up to 13 dimensions), the AE is capable of providing a representation of the TS SOH/RUL in a smaller-dimensional latent space (2D for the TrajecNets approach). Indeed, the RUL is estimated based on the smaller-dimensional representation that stems from the computation of current and past data. Due to its complexity, this model may involve more than 30,000 weights to be trained. During training, the resulting TrajecNets' RNN learns the neurons' weights with the objective of minimizing the reconstruction error of the AE and the exponentially weighted mean absolute error of the estimated RUL. For more details about the training procedure, and inner details of TrajecNets, please refer to (Shahid & Ghosh, 2019). The described CBM framework has been implemented using the widely adopted TensorFlow library, and it has been previously validated using publicly available datasets (Shahid et al., 2019).

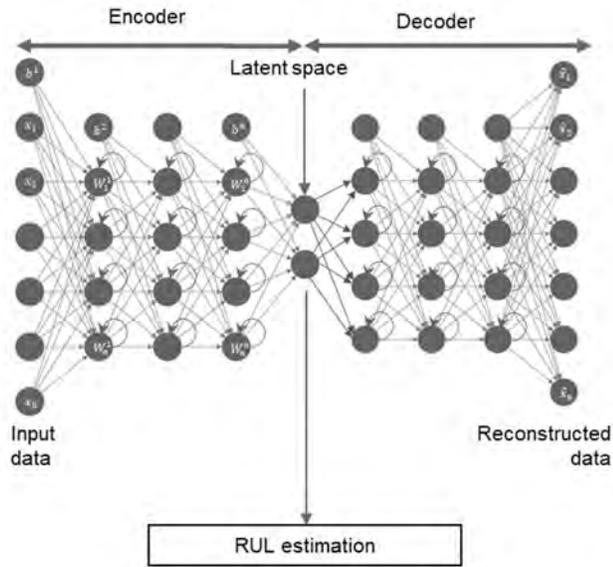


Figure 6 – TrajecNets architecture for data-driven RUL estimation (Shahid & Ghosh, 2019).

3.2. Mapping to the reference architecture

The implementation of the CBM framework described in section 3.1 requires support for the following high-level functions:

- DA: the input data to the TrajecNets RNN must be collected from the TS.
- DM: the proposed algorithm assumes that the input data are collected during specific flight phases, with different pre-processing requirements. A set of relevant features must be extracted from the acquired time-series.
- HA and PA: as described, the data-driven model developed and trained using the collected data can support in providing TS’s HA and RUL estimation.

The SD function is not considered in the context of this work, but it can be potentially integrated in the CSM.

The DA function shall be allocated both on the TS and on the CSM, according to the strategies already detailed in section 2.1. Considering the low computational complexity, it is possible to deploy the DM functions directly on the CSM. One possible limitation related to the feature extraction could be related to the need of storing the acquired time series on CSM’s memory, which can be in general limited. Implementing online calculation of target features (e.g., statistical indices) may help in mitigating this risk.

As discussed, the integration of the prognostics function on the on-board CSM represents an interesting, but challenging, aspect that requires a reduction in size for the TrajecNets-based RUL model in order to fit the constrained resources of the on-board system.

There are different approaches to reduce the size of a model, such as quantization, pruning and clustering. Quantization

can reduce the size of a model by mapping continuous variables onto discrete values, potentially at the expense of some accuracy (such as truncating or rounding). Pruning and clustering can reduce the size of a model by making it more compressible. Pruning works by removing parameters within a model that have only a minor impact on its predictions. Pruned models are the same size on disk, and have the same runtime latency, but can be compressed more efficiently. Clustering works by grouping the weights of each layer in a model into a predefined number of clusters, then sharing the centroid values for the weights belonging to each individual cluster. This reduces the number of unique weight values in a model, thus reducing its complexity. Clustered models can also be compressed more effectively, providing some deployment benefits.

As anticipated, the TrajecNets-based RUL model developed in this work relied on the usage of the TensorFlow framework; for this reason, the TensorFlow Lite toolset (Google, 2022) appeared to be a natural option to support the RUL’s model size reduction.

3.3. Edge platform selection

Different classes of devices can be considered for the implementation of the on-board CSM, ranging from microcontroller devices to more capable single board computers. Considering the need to host middleware-related functionalities, and potentially supporting advanced frameworks like TensorFlow, a mid-range single board computer has been identified.

Specifically, the target platform considered in this work is a commercial single-board computer based on a Freescale i.MX6 System on Chip (SoC), including an Arm® Cortex®-A9 single core, operating at a clock frequency of 800 MHz. The SoC also include a Graphics Processing Unit (GPU), providing hardware acceleration for 3D graphics but not suitable for more general-purpose computation. The platform manufacturer provides support for several GNU/Linux OS distributions. The TensorFlow framework has been fine-tuned to run on top of the standard Ubuntu Linux distribution. The platform has a base storage capability of 4GB, with support for external expansion. It supports a variety of communication interfaces, including wired and wireless networking.

3.3.1. Software architecture overview

The proposed hardware platform provides a variety of I/O interfaces, suitable for interfacing with the TS and for communicating over an airborne network supporting the overall PHM functions.

The software architecture of the proposed system is represented in Figure 7.

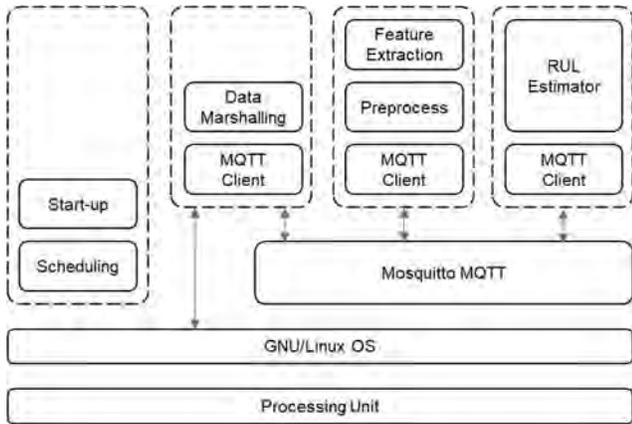


Figure 7 – Proposed CSM software architecture

The DA component supports communication over external interfaces, data marshalling, and interaction over the MQTT middleware, which, as anticipated, provides an abstraction for network communications.

The feature extraction component supports the online calculation of target features set and handles the interactions with the MQTT middleware. The component receives as input the information on the current flight phase and relies on it to apply the correct pre-processing approach to the input data.

Finally, the CSM supports PA and HA, based on the algorithm described in section 3.1. This component relies on the TensorFlow Lite runtime and integrates an MQTT client. Besides the described function, the platform includes a standard generic component supporting the initial start-up of the platform (initialization of networking services and of MQTT middleware support) and scheduling of other components.

4. VALIDATION SETUP: SUPPORTING CBM FOR AN AIRCRAFT BRAKING SYSTEM

To validate the on-board PHM architecture introduced in the previous sections, in the following part of the paper, a representative braking system is presented as TS.

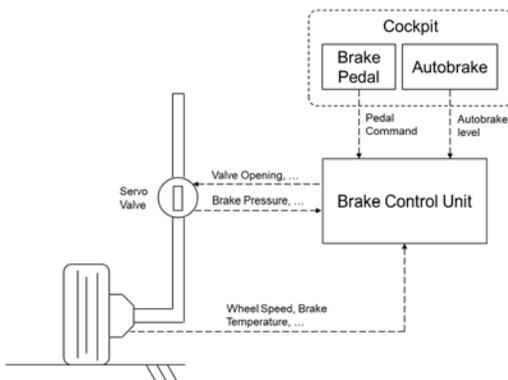


Figure 8 - High-level view of an Aircraft Braking System

Figure 8 shows a simplified control architecture for an aircraft electro-hydraulic/electro-hydrostatic braking system (SAE International, 2006).

Common commercial aircraft brakes are made of Carbon/Carbon composites. The brake is composed of a stack of alternated stator and rotor discs. A servo valve regulates the hydraulic fluid directed to the pistons in the brake assembly. An increase of the hydraulic pressure causes a compression in the stack of stators and rotors, resulting in an increase of friction that slows down the wheel rotation. During the braking action, most of the aircraft’s kinetic energy is transformed into heat and absorbed by the brakes (Daidzic, 2017).

A Brake Control Unit (BCU) hosting the control system is responsible for the regulation of the servo valve. The valve opening is computed according to the reference signals received from the cockpit, namely the pilot (and/or co-pilot) brake pedal command and/or the level of required autobrake. The BCU can collect data from several sensors deployed along the system (wheel speed, brake temperature, brake pressure, etc.), and uses them as input for the different control algorithms hosted on the unit.

After each brake application, the brake’s discs tend to wear (due to the loss of material induced by friction). Moreover, wear dynamics are impacted by the thermal behavior of the brake (Di Santo, 2005). When the level of wear reaches certain limits, the brake needs to be replaced with a new one. It is therefore possible to relate a brake’s wear to its RUL.

The following subsections provide an overview of a possible application of the generic framework described in section 3 for brake RUL estimation, along with its actual deployment of an on-board setup that follows the architectural approach presented in section 2.1. Due to the confidentiality of the data presented, normalization procedures were carried out. Despite the normalization process, the results presented in the following section highlights the efficacy of the proposed approach in supporting the scalable deployment of complex PHM functions onto on-board components with limited resources.

4.1. Data generation, acquisition, and manipulation

To generate data and support the training phase of the TrajecNets-based RUL estimator for brakes, a MATLAB-based simulator was built to emulate the behaviour of both aircraft brakes’ wear and thermal dynamics.

A representative set of variables collected from the simulator were used to train TrajecNets (example subset shown in Table 1).

The simulated data are sampled with a frequency of 1 Hz and pre-processed applying a moving average. The acquisition and pre-processing of input data leads to a total amount of 1820 data items per flight, for a total size of ~7 KBs.

Table 1 – Representative set of input variables

Variable	Description
Wear level	Indicator that provides a quantification of the current brake’s wear level.
Brake temperature	Indicator that provides the current temperature of the brake.
Flight phase	Flag that indicates in which flight phase the system is working in.

As discussed in section 3.1, a set of features was computed to support the brake RUL estimation. These features are computed for each input variable, per each flight phase of a given flight. It should be noted the features could be computed online, leading to a potential further reduction of memory footprint.

The outcome of the training step provided a TrajecNets-based model (original model) trained with a high-fidelity parameterization of its internal structures (~30,000 32bit float parameters). However, due to its complexity and dimensionality, the online execution of such model onto the edge computing device defined in section 3.3 resulted to be prohibitive.

4.2. Model reduction, integration & assessment

To overcome the edge deployment limitations, a model reduction and integration process was carried out. The original model was reduced to a quantized version through TensorFlow Lite, where a quantization of the ~30,000 parameters from float (32bits) to integer (16bits) has been adopted.

The overall reduction process required two steps:

- (i) the original model with a size of 340 KBs, was initially *reduced* to a TensorFlow Lite float model of 211 KBs in size, and
- (ii) the float model was finally *reduced* into the quantized version which size was 164 KBs (about 77% of the float model size).

The impact of the model size reduction through quantization on the model accuracy has been evaluated through simulation. The accuracy of the model is defined as the Root Mean Squared Error (RMSE) between the RUL prediction and a target value, over a statistically meaningful number of unseen simulated flights.

From the comparison results, it emerged that both the quantized and float models provided RMSE indices ~5% higher with respect to the original model (considered our best-case scenario). The accuracy of the reduced models is slightly impacted but not in a significant manner for the application under analysis.

The performance of the reduced models has been characterized on the reference platform using the standard

TensorFlow Lite Benchmark tool. The tools allow to measure relevant execution times (initialization time, inference time of warmup state, inference time of steady state) and memory usage (memory usage at initialization time, overall memory usage during execution). As a reference, the same characterization for the reduced models executed on a high-end processor (Intel Core i5-8365U with a clock frequency of 1.60 GHz) is included.

The inference timings of the reduced models are summarized in Figure 9 and Figure 10, respectively for the quantized version and the float version.

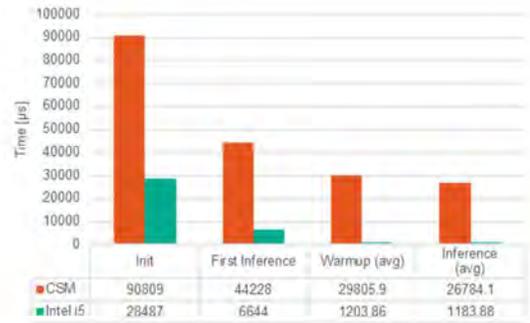


Figure 9 – Inference times, quantized model

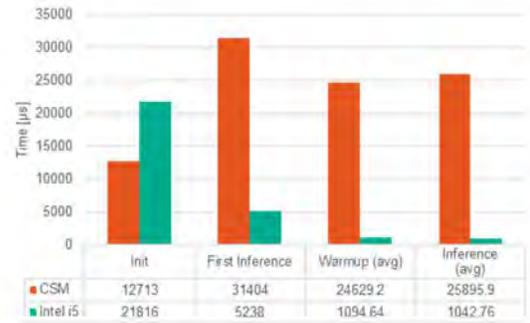


Figure 10 – Inference times, float model

In both cases, estimated time figures resulted acceptable with respect to target performance requirements, especially considering the non-real time nature of the execution for this part of the application.

The peak memory footprint for the reduced models is detailed in Table 2 and Table 3, respectively for the quantized version and the float version.

Table 2 - Quantized model, peak memory footprint [MB]

	CSM	Intel i5
Init	2.74	3.24
Overall	3.52	3.99

Table 3 - Float model, peak memory footprint [MB]

	CSM	Intel i5
Init	2.93	4.16
Overall	3.64	4.78

Both in the case of the quantized model and the float model, the peak memory footprint proved to be compatible with the memory capabilities of the described processing platform.

5. CONCLUSION

This paper presented a strategy for the scalable deployment of PHM techniques for on-board systems, with particular focus on edge computing capabilities.

The relevant scenario for deployment of CBM support for on-board systems have been presented, and a system level architecture able to address significant use-cases has been introduced. The proposed approach allows flexibility both in the on-board deployment and in targeting different on-board systems.

In the proposed approach, the TS is extended with an external support module, able to host relevant CBM related functions.

The application of the proposed approach to an actual aerospace use case has been discussed, first introducing a generic CBM framework, previously validated in a traditional offline setup, and then mapping this framework over the proposed reference architecture.

The proposed implementation proved to be viable in a real use case, a representative aircraft braking system, used for the overall validation of the proposed approach. The described methodology is in any case generalizable and can be applied to different aircraft systems.

Based on proposed analysis, it was shown that the TrajecNets-based model trained off-line using simulation data (and based on the full TensorFlow library) can be converted and reduced in size (and complexity) for an embedded application using TensorFlow Lite toolset. A non-significant impact on the model accuracy for RUL estimation related to brake wear was evaluated for this use case.

ACKNOWLEDGEMENT

This research has been funded by the European Union’s CleanSky2 Research and Innovation program ICO-Brake under grant No. 945535. The authors would like to thank Jeffrey Schmidt, Rhonda Walthall, and Marc Georjin for their support during the execution of the activities described in this paper.

REFERENCES

- Brady, C. (2022). Brake Wear Pin. Retrieved from B737: <http://www.b737.org.uk/brakepin.htm>
- Cadini, F., Zio, E., & Avram, D. (2009). Model-based Monte Carlo state estimation for condition-based component replacement. *Reliability Engineering & System Safety*, 94 (3), pp. 752–758.
- Chang, S., Gao, L., & Wang, Y. (2018). A review of integrated vehicle health management and prognostics and health management standards. *International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC)*, (pp. 476-481)
- Chen, R., Hu, Y., & Hou, Y. (2021). Architecture analysis for avionics prognostics and health management system. *CAA Symposium on Fault Detection, Supervision, and Safety for Technical Processes (SAFEPROCESS)*, (pp. 1-5)
- Chen, Z., Liu, J., & Zhou, X. (2020). An Embedded AI Computing Platform for Aircrafts Based on VPX Bus. *4th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, (pp. 602-606)
- Daidzic, N. E. (2017). Modeling and Computation of the Maximum Braking Energy Speed for Transport Category Airplanes. *Journal of Aviation Technology and Engineering*, 2-25
- Di Santo, G. (2005). Proper Operation of Carbon Brakes. *11th Performance and Operation Conference*
- ISO. (2003). Condition monitoring and diagnostics of machines - Data processing, communication and presentation - Part 1: General guidelines, ISO 13374-1:2003
- Ezhilarasu, C. M., & Jennions, I. K. (2021). Development and Implementation of a Framework for Aerospace Vehicle Reasoning (FAVER). *IEEE Access*, 9, 108028-108048.
- Finda, J., & Hédl, R. (2014). On-board SHM System Architecture and Operational Concept for Small Commuter Aircraft. *PHM Society European Conference*, 2 (1)
- Goebel, K., & Rajamani, R. (2021). Policy, regulations and standards in prognostics and health management. *International Journal of Prognostics and Health Management*, 12(1).
- Google LLC. (2022). TensorFlow Lite: Deploy machine learning models on mobile and IoT devices. Retrieved from <https://www.tensorflow.org/lite/>
- Han, D. Y. (2019). A distributed autonomic logistics system with parallel-computing diagnostic algorithm for aircrafts. *2019 IEEE AUTOTESTCON*, (pp. 1-8)
- Li, R., Verhagen, W. J., & Curran, R. (2020). A systematic methodology for Prognostic and Health Management system architecture definition. *Reliability Engineering & System Safety*.
- MIMOSA. (2022). Open System Architecture for Condition Based Monitoring (OSA-CBM). Retrieved from: <https://www.mimosa.org/mimosa-osa-cbm/>

- Mosallam, A., Medjaher, K., & Zerhouni, N. (2015). Component based data-driven prognostics for complex systems: Methodology and applications. *First International Conference on Reliability Systems Engineering*, (pp. 1–7)
- Reddy, K. K., Venugopalan, V., & Giering, M. J. (2016). Applying deep learning for prognostic health monitoring of aerospace and building systems. *1st ACM SIGKDD Workshop on ML for PHM*
- SAE International. (2006). Braking System Dynamics, AIR1064D
- SAE International. (2018). Design & Run-Time Information Exchange for Health-Ready Components, SAE JA6268
- Shahid, N., & Ghosh, A. (2019). TrajecNets: Online failure evolution analysis in 2D space. *International Journal of Prognostics and Health Management*, 10(4)
- Tambe, S. U. (2013). An extensible architecture for avionics sensor health assessment using data distribution service. *AIAA Infotech@Aerospace (I@A) Conference*, (p. 5139)
- Terrissa, L. S., Meraghni, S., Bouzidi, Z., & Zerhouni, N. (2016). A new approach of PHM as a service in cloud computing. *4th IEEE International Colloquium on Information Science and Technology (CiSt)*, (pp. 610-614)
- Wang, F., Pan, S., Xiong, Y., Fang, H., & Wang, D. (2017). Research on software architecture of prognostics and health management system for civil aircraft. *International Conference on Sensing, Diagnostics, Prognostics, and Control*
- Wang, T. (2010). *Trajectory similarity-based prediction for remaining useful life estimation*. Doctoral dissertation, University of Cincinnati.
- Zhang, A., Wang, H., Li, S., Cui, Y., Liu, Z., Yang, G., & Hu, J. (2018). Transfer learning with deep recurrent neural networks for remaining useful life estimation. *Applied Sciences*, 8 (12), p. 2416
- Technologies Corporation (now part of Raytheon Technologies) in 2015 as Senior Engineer, working for Collins, in the Applied Research and Technology organization. She was member of the Embedded Technologies group, now she is member of the Digital Thread & Twin Technologies.

Mathieu Le Cam is a senior researcher with Collins Aerospace Ireland. He received his PhD in Building Engineering from Concordia University, Canada in 2016 and his M. Eng. in Mechanical Engineering from Ecole Polytechnique de Montreal in 2012. His academic research focused on data-driven and physics-based modeling for the forecast of the electric demand of HVAC systems, and their application in building energy management. Since he joined Collins Aerospace Ireland, formerly United Technologies Research Centre, in 2017, he has been involved in different projects leveraging his skills in machine learning and mechanical engineering for the Prognostics and Health Managements of different aircraft systems.

Marcello Torchio obtained the master’s degree in computer engineering from the University of Pavia in 2012. In 2016 he received the PhD in Automatic Controls at the University of Pavia with the Thesis entitled “Model Predictive Control Strategies for Advanced Battery Management Systems”. In 2016 he joined United Technologies Research Centre Ireland, ltd as a Senior Research Scientist in the Controls and Decision Support group providing technical and management contributions in the execution of R&D projects. From 2021 he covers the role of Sr. Principal Engineer at the Advanced Research & Technology center of Collins Aerospace, within the Autonomous Systems team. He is author of several international journal and conference papers on topics of model predictive control, optimization, and electrochemical modelling, as well as author of different patents.

David Larsen is a Technical Fellow, PHM Systems at Collins Aerospace. Dave is a leader in the advancement of Prognostics and Health Management (PHM) Systems for Collins, authoring Collins’ common PHM standard work methods, driving PHM collaboration, and leading the design and development of PHM-enabling systems and components, specializing in health data provisioning. Dave teaches the Introduction to PHM course and the PHM Fundamentals and Use Cases course through the Collins Aerospace Technical University. Dave is also the Dean of the CATU Systems Engineering College. Dave holds a BSEE from Rensselaer Polytechnic Institute. Dave is a Fellow of the PHM Society.

BIOGRAPHIES

Fabio Federici is a Principal Engineer at Collins Aerospace, a Raytheon Technologies company. At Collins, he works as part of the Applied Research and Technology organization, mainly focusing on dependable embedded HW/SW architectures. He originally joined the United Technologies Corporation (now part of Raytheon Technologies) in 2017, as part of the corporate research center. Before that, he worked as a Data Handling and Research Engineer at Thales Alenia Space Italy. Fabio holds a Ph.D. in Electrical and Information Engineering from the University of L’Aquila (Italy). He has been a post-doctoral researcher and a contract professor at the same university.

Cecilia Tonelli earned a M.S. in Mathematics, with a Computational Algebra thesis in 2011 and one year later she participated the 2012 EACA (Meetings on Computer Algebra and Applications). She worked as Junior Software Developer in automotive sector for 3 years, then she joined the United

Sensor fault/failure correction and missing sensor replacement for enhanced real-time gas turbine diagnostics

Amare Fentaye¹, Valentina Zaccaria², and Konstantinos Kyprianidis³

^{1,2,3}*Future Energy Center, Mälardalen University, 72123 Västerås, Sweden*

amare.desalegn.fentaye@mdu.se

valentina.zaccaria@mdu.se

konstantinos.kyprianidis@mdu.se

ABSTRACT

Gas turbine sensors are prone to bias and drift. They may also become unavailable due to maintenance activities or failure through time. It is, therefore, important to correct faulty signal or replace missing sensors with estimated values for improved diagnostic solutions. Coping with a small number of sensors is the most difficult to achieve since this often leads to underdetermined and indistinguishable diagnostic problems in multiple fault scenarios. On the other hand, installing additional sensors has been a controversial issue from cost and weight perspectives. Gas path locations with difficult conditions to install sensors is also among other sensor installation related challenges. This paper proposes a sensor fault/failure correction and missing sensor replacement method. Auto-regressive integrated moving average models are employed to correct measurements from faulty and failed sensors. To replace additional sensors needed for further diagnostic accuracy improvements, neural network models are devised. The performance of the developed approach is demonstrated by applying to a three-shaft turbofan engine. Test results verify that the method proposed can well-recover measurements from faulty/failed sensors, no matter with small or major failures. It can also compensate key missing temperature and pressure measurements on the gas path based on the data from other available sensors.

Keywords: Gas turbine sensors; sensor fault; sensor failure; signal reconstruction; missing sensor replacement

1. INTRODUCTION

The quality and quantity of performance data collected along the gas path is key for an accurate gas turbine diagnostics. This depends on the number and type of sensors installed in

real-life. In principle, gas path sensors should preferably be placed in the entry and exit of the critical gas path components, to get the complete picture of the engine health. However, this is not often possible in real situations for several reasons. The major sensor related challenges from the diagnostic perspective are discussed as follows.

The first challenge is measurement noise. Noise affects early fault detection ability by hiding low-level fault signatures. It also increases false alarms during harsh operating conditions. Additionally, noise is known to cause Smearing effects in physics driven diagnostic methods (A. Fentaye, Zaccaria, & Kyprianidis, 2021; Zaccaria, Fentaye, Stenfelt, & Kyprianidis, 2020). Data denoising prior to fault diagnostic activities (Y. G. Li, 2002; Sadough Vanini, Meskin, & Khorasani, 2014) and developing noise tolerant diagnostic methods (Bettocchi, Pinelli, Spina, & Venturini, 2006) are the two widely studied solutions.

A sensor fault/failure is the second challenge (Jombo, Zhang, Griffiths, & Latimer, 2018). Bias and drift are the two known forms of a sensor fault. Bias is a systematic measurement error which results in fixed and abrupt shifts. Installation errors, high vibrations and harsh working conditions can be the root causes. Drift is the other source of inaccurate measurements associated with sensor age. In some catastrophic working environments, a complete sensor failure is also expected. This includes not receiving signals and stuck to some specific readings. To diagnose an engine with a sensor fault/failure occurrence, either the faulty signals should be corrected first, or the diagnostic system should be tolerant enough to the corrupted data (J. Li & Ying, 2020; Lu, Li, Huang, & Jia, 2020). Another alternative is separating sensor faults from component faults before conducting any further fault analysis (Ogaji & Singh, 2003).

The third challenge is that certain sensors may become unavailable through time due to maintenance activities or hostile operating conditions. This will lead to fleet engines having different gas path sensors, particularly between the older engines and brand-new ones. Consequently, a life-cycle

Amare Fentaye et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

performance monitoring and diagnostic system needs to be modified to cope with the available sensors, or the missing sensor must be installed or automatically replaced by estimated values from the remaining measurements. Modifying the performance monitoring and diagnostic system to cope with the remaining sensors may be difficult, since this may cause undistinguishable failure modes, underdetermined diagnostic problem, or may have a significant impact on the diagnostic accuracy. The second alternative is to automatically replace the missing sensor by estimated values from redundant and other available measurements. The estimation can be performed based on physics (Aivaliotis, Georgoulas, Arkouli, & Makris, 2019; X. Zhou, Lu, & Huang, 2019) as well as using machine learning methods (Kramer, 1992).

Sensors can also be missing due to absence of technology. For instance, high-pressure turbine (HPT) entry and exit temperature and pressure sensors are unavailable due to the high gas temperature. Nevertheless, these measurements are vital to discriminate failure modes between the gas turbine hot components (A. Fentaye, Zaccaria, Rahman, Stenfelt, & Kyprianidis, 2020; Zaccaria et al., 2020; X. Zhou et al., 2019). It is technically possible to measure HPT and intermediate-pressure turbine (IPT) blade metal temperatures using a pyrometer. For instance, the RB199 and EJ200 military turbofan engines have a pyrometer on the HPT. But this is not available yet for civil aeroengines and does not represent the actual gas path temperature either.

Cost and weight reduction is another reason why some configurations do not include some measurements. Historically, gas turbine sensors are installed primarily for safety and control purposes. Mostly they are equipped with less than 100 sensors. Modern gas turbines, in contrast, are equipped with over 5000 sensors in total. (Afman, Prasad, & Antolovich, 2017) This includes additional gas path temperature and pressure probes installed for diagnostic reasons. However, several studies on gas turbine diagnostics such as (Ganguli, 2002; Jasmani, Li, & Ariffin, 2011; Simon & Rinehart, 2016) indicated that even the modern gas turbines are still missing some useful gas path measurements which could potentially improve the diagnostic accuracy considerably. A measurement selection study conducted for the Rolls-Royce RB211-24G showed that the high-pressure compressor (HPC) exit temperature (T5), HPT exit pressure (P9), and power turbine (PT) exit temperature (T12) measurements are among the most critical measurements to accurately diagnose the engine gas path (Jasmani et al., 2011). However, all these sensors are not included in the installation. Hence, due to the trade-off between cost plus weight reduction and improving diagnostic accuracy, the decision of installing more sensors is controversial.

Redundancy between measurements (Jasmani et al., 2011) and singularity (Kaboukos, Oikonomou, Stamatis, & Mathioudakis, 2003) are additional challenges of sensors for

fault diagnostics. Since sensors are basically installed for the sake of control and safety, some of them may not be useful for diagnostics due to redundancy and singularity issues. Redundancy of measurements is the phenomenon of having two or more sensors with high correlation, while singularity is the state of sensors not responding to performance changes.

Lack of sufficient measurements due to the above highlighted reasons highly affects the maintenance decision making process. For model-based diagnostic approaches, the common way to deal with underdetermined and undistinguishable problems is to choose the best subset of the performance parameters that substantially represent the engine condition (Kaboukos et al., 2003). However, this technique is not effective enough since changes from the removed performance parameters propagate to the selected ones (Simon & Garg, 2009). Two measures can be taken to overcome the shortcoming: installing additional sensors or replacing the missing sensors with estimates. The former had not been of interest to both engine manufacturers and end users due to sensor related costs. Using models as virtual sensors has become indispensable in modern industry for process control, online monitoring, and diagnostics (Jiang, Yin, Dong, & Kaynak, 2021; Kamat & Madhavan, 2016). Recently, this topic has been also receiving attention by the gas turbine community (Afman et al., 2017; J. Zhou, Liu, & Zhang, 2016).

This paper aims to explore the use of autoregressive integrated moving average (ARIMA) models to correct measurements from faulty sensors as well as failed sensors until recalibration or reinstallation is performed. Two modes of operation are considered: when a sensor fault/failure occurs while the engine is under the normal degradation mode and when a sensor fault/failure simultaneously occurs with a component fault. Neural network (NN) models are employed to replace missing sensors due to maintenance activities and additional sensors required to improve diagnostic accuracy.

The rest of the paper is organized as follows. Section 2 describes the method proposed and the case studies used to demonstrate and validate the method. The implementation results to a test case aeroengine with detailed discussion is presented in Section 3, followed by key concluding remarks.

2. METHOD

A generic framework shown in Figure 1 is proposed to address the above discussed sensor problems. From the diagnostic perspective, the first step is selecting the best measurement set that can allow accurate gas path analysis (GPA). Measurements are selected through a sensitivity and correlation analysis. Even among those limited measurements, few of them should be removed, if they are highly correlated with other measurements and/or insensitive to key performance deviations. Various combinations of measurements can also be needed depending on power setting parameters and flight conditions considered.

One of the widely applied methods to deal with limited sensors is choosing the subset providing maximum accuracy. For an invertible system matrix, the number of performance parameters should be reduced to the number of measurements selected. Checking all the possible performance parameter combinations and selecting the subset with maximum accuracy is an iterative process. This approach is prone to error propagation problem due to the unestimated performance parameters and time consuming. Observability analysis can help reduce some of the performance parameters showing high correlation. All the remaining gas path faults should be easily isolable by interpreting the measurement deviations. Choosing among the highly correlated performance parameters to be discarded could be decided based on user’s priorities (Stenfelt & Kyrianiadis, 2022).

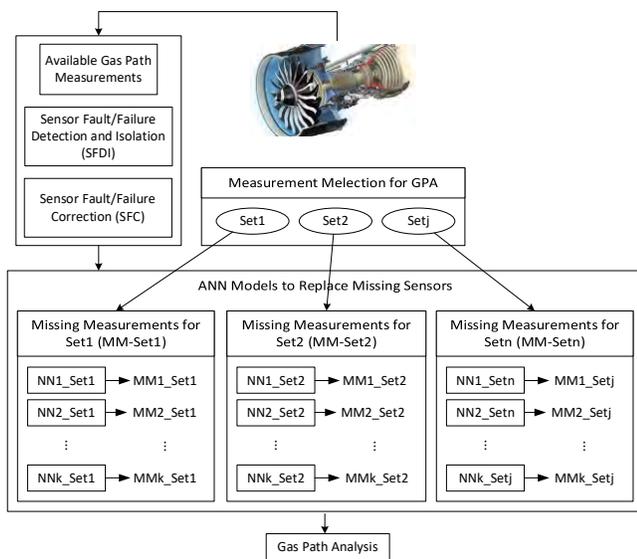


Figure 1. Proposed framework.

In this proposed framework, Figure 1, a full set of health parameters associated with critical gas path components were considered. The necessary measurements were selected through a sensitivity and correlation analysis using performance model of the engine. Unavailable physical measurements are replaced by independently acting NN models developed based on the available information. One important point to note here is that, unlike for diagnostics, if two or more measurements among the available ones are showing high correlation, none of them should be discarded. They are important for each other’s predictions during failure.

Since some sensors could become unavailable through time due to maintenance events and hostile operating conditions, the physical sensors available could vary with the engine age. This decreases the number of input measurements to the NN models, while increasing the number of NN modules required to replace missing sensors. Suppose the engine is equipped with m sensors when it was brand new and n measurements are selected for accurate invertible diagnostics, m-n NN

modules will be needed to replace the missing sensors. To account missing sensors through time, the NN method should be evaluated for fewer instrumentation suites as well.

In addition to missing sensor cases, a sensor can start providing false readings (due to bias or drift faults) or fixed readings (due to a complete failure) as illustrated in Figure 2. To continue the plant operation, process monitoring, and diagnostic tasks with no interruption, those readings should be automatically replaced by estimated values. In this paper, a forecasting approach was applied to recover inaccurate signals due to a sensor fault/failure. Actual measurements after the detection point are estimated based on the principle of time-series forecasting, by projecting the values before the sensor fault/failure has been detected. The popular forecasting method, ARIMA, is used. However, in a complete engine health monitoring system, this step should be preceded by a sensor fault detection and isolation (SFDI) process, which is not the scope of this paper.

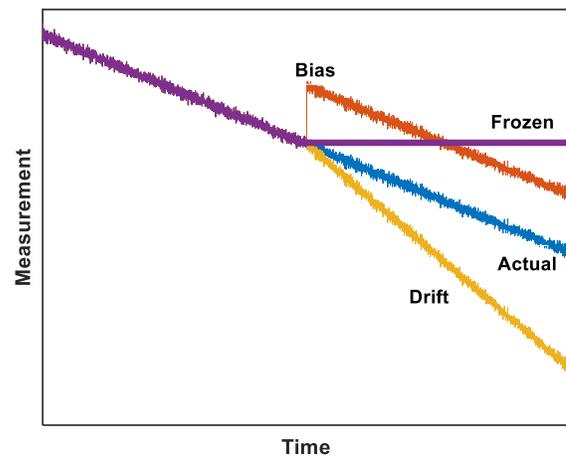


Figure 2. Schematic illustration of a sensor fault/failure in a deteriorating engine.

2.1. Sensor Fault/Failure Correction using ARIMA

The Box-Jenkins ARIMA is a statistical technique widely used for time-series forecasting. It is often designated as ARIMA(p,d,q). and consists of three key parameters: Autoregression (AR), Integrated (I) and Moving Average (MA). The AR part fits a time-series data and forecast future values based on previous values. The MA term uses past errors to make future predictions. The “I” term is needed to make the time-series stationary by taking differences, with d order of transformation. The underlying mathematics can be expressed as

$$y_t = \alpha + \gamma_1 y_{t-1} + \gamma_2 y_{t-2} + \dots + \gamma_p y_{t-p} + \varepsilon_t + \beta_1 \varepsilon_{t-1} + \beta_2 \varepsilon_{t-2} + \dots + \beta_q \varepsilon_{t-q} \quad (1)$$

where y_t is the actual value at time t and ε_t is the associated normally distributed random error, α and γ represent model parameters, p and q are integer values that indicate orders of

the AR and MA models, respectively. $q=0 \Rightarrow$ AR model of order p , and $p=0 \Rightarrow$ MA model of order q .

Four main steps are required to develop an ARIMA model: model identification, parameter estimation, model validation, and forecasting. The first step is used to determine the p , d , and q values from the sample data. It also involves converting the series to stationary through differencing. In the second step, α and γ are computed to optimize the overall error between the predicted and actual input-output data. Least-squares method or other nonlinear optimization techniques can be applied here. Once these two steps are completed, the accuracy of the ARIMA model is verified using new data. The forecasting step is then followed for future timesteps.

ARIMA models have been widely used for engine RUL forecasting so far (Marinai, Singh, Curnock, & Probert, 2003). In our proposed framework, ARIMA is used to reconstruct measurement errors induced by a sensor fault or failure. This is important to avoid unnecessary downtimes for sensor maintenance. Particularly, if the problematic sensor is in the control loop, the reconstructed values can automatically be used for that period to continue the system process without any interruption plus estimate the magnitude of the engine deterioration. Regardless of knowing the type of sensor problem and its magnitude, the ARIMA model forecasts the actual readings based on the readings collected before the problem has been detected. Using the available fault-free signals from the same sensor and correcting the error part through forecasting is more effective than estimating the values based on measurements from other fault free sensors.

However, if a sensor bias is detected due to installation errors, the ARIMA model cannot be applied. Instead, the signals from this sensor should be replaced by a NN model estimates until a reinstallation or recalibration measure is taken. Moreover, if a component fault occurs after a sensor failure, the failed sensor readings should be replaced with estimated values from the NN module. Because the component fault induced measurements cannot be predicted by the ARIMA model based on the normal data or trend before the sensor failure.

2.2. Missing Sensor Replacement using NNs

The typical feed-forward multilayer perceptron has a proven performance record to learn relationships between variables from data and make accurate predictions. In the method proposed, independently acting NN models are developed for each sensor as a regression problem to replace when they are missing, and to replace other additional sensors required for enhanced diagnostic solutions. All available measurements were used as input to each network. The output is the estimated version of a weighted sum of the input.

Model hyperparameters for each NN estimator were determined through a training process the Levenberg-Marquardt backpropagation algorithm. Although a NN model with one hidden layer, with sufficient neurons, is known to be capable enough to approximate most nonlinear regression problems, different number of hidden layers (up to 3) and neurons (up to 60) were evaluated. Variety of activation functions (tanh, logsig, and ReLU) and optimization algorithms (Adam, RMSProp, and sgd) with different data structure and number of epochs were also checked. The most appropriate network structure was then selected based on the training time, accuracy, and robustness. The Adam optimization algorithm and ReLU activation function showed better performance. The accuracy of the predictions was assessed using the standard deviation (σ) mean absolute error (MAE) and root mean square error (RMSE) indicators.

2.3. Case study: three-shaft turbofan engines

The schematic of the case study engine is shown in Figure 3. Full set of performance parameters, i.e., efficiency and flow capacity for the Fan, intermediate-pressure compressor (IPC), HPC, HPT, intermediate-pressure turbine (IPT), and low-pressure turbine (LPT) were considered. Table 1 presents selected measurements via an observability analysis using the engine performance model and the measurement noise considered. The noise incorporated was a normally distributed Gaussian noise with zero-mean and standard deviation varies as percentage of the actual sensed values.

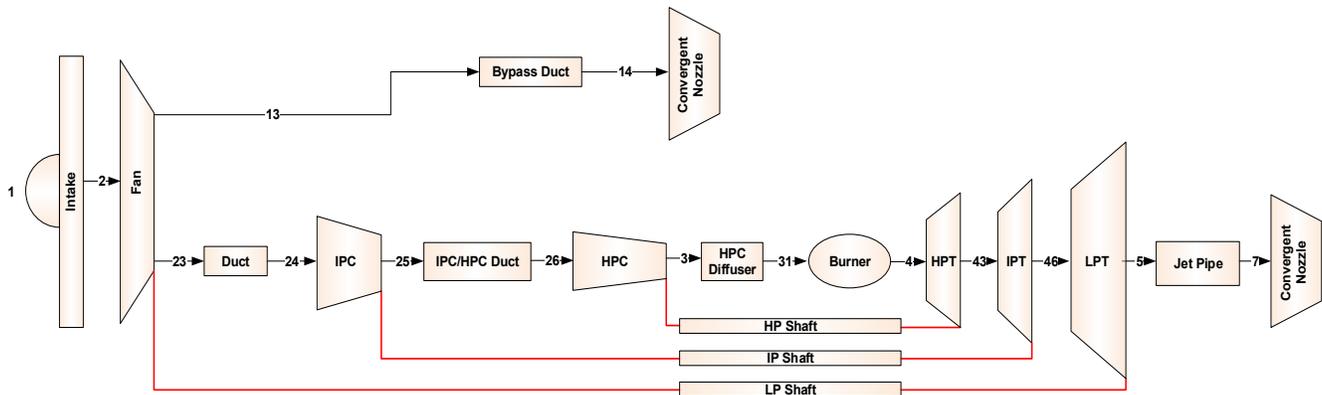


Figure 3. Schematics of three-shaft turbofan engine with measurement locations.

Table 1. Selected measurements for 3-shaft turbofan engine.

Measurement parameter	Notation	Unit	Noise (σ)
IPC inlet total temperature	T23	K	$\pm 0.4\%$
IPC inlet total pressure	P23	kPa	$\pm 0.25\%$
IPC exit total temperature	T25	K	$\pm 0.4\%$
IPC exit total pressure	P25	kPa	$\pm 0.25\%$
HPC exit total temperature	T3	K	$\pm 0.4\%$
HPC exit total pressure	P3	kPa	$\pm 0.25\%$
HP shaft speed	N4	rpm	$\pm 0.05\%$
IP shaft speed	N43	rpm	$\pm 0.05\%$
LP shaft speed	N46	rpm	$\pm 0.05\%$
HPT exit total pressure	P43	kPa	$\pm 0.25\%$
IPT exit total pressure	P46	kPa	$\pm 0.25\%$
LPT exit total temperature	T5	kPa	$\pm 0.25\%$

Based on (Marinai, 2004; Saias, Pellegrini, Brown, & Pachidis, 2021), N4, N44, N46, T25, P25, T3, P3, T46, and T5 are considered mostly available sensors for the commercial 3-shaft turbofan engine. However, sensitivity and correlation analysis results indicate that, N4, N44, N46, T23, P23, T25, P25, T3, P3, P43, P46, and T5 are important to distinguish and estimate the full set of health parameters of the turbofan gas path. Although sensors that have strong correlation between them are not needed in the main diagnostic scheme, they are useful to estimate each other when they fail. That is why high correlation sensors are included in the proposed method. To address the sensor fault/failure correction and missing sensor replacement problems, the following cases were investigated:

1. A single sensor fault/failure was considered and independently acting ARIMA models devised for each available sensor for correction. The engine was under gradual degradation mode.
2. Sensor fault/failure correction was carried out when it simultaneously occurs with a component fault. It is less likely to occur both a sensor fault/failure and a component fault together, but for the sake of inclusiveness this case was also investigated.
3. Four independent NN modules were devised to replace the missing T23, P23, P43, and P46 measurements.
4. Six other NN estimators were developed for the available measurements T25, P25, T3, P3, T46 and T5 to replace them when they are missing.
5. The sensitivity of T23, P23, P43, and P46 estimators was also analysed when case 4 happens.

Data required for training and testing was generated via model simulation. In-house software EVA was used for the simulation as described in (Kyprianidis, 2017). A database of 127080 random inputs was generated considering healthy, deteriorated, and faulty engine conditions. For the sake of generic estimation models, the fault patterns were derived from 63 single and multiple fault types, obtained by considering all possible combinations of the six gas path components taken r each time (where $r = 1, 2, \dots, 6$). The

fault magnitude (FM) was a function of efficiency and flow capacity changes ($\Delta\Gamma:\Delta\eta$) as expressed in Equation (2). For the Fan and the two compressors, equal fault magnitudes (up to 5%) were considered. To accommodate possible ratio differences between efficiency and flow capacity performance factors (A. D. Fentaye, Baheta, Gilani, & Kyprianidis, 2019), 12 different ratios (from 1:1 to 4:1) were considered. Likewise, for the three turbines, equal fault magnitudes (up to 4%) with similar $\Delta\Gamma:\Delta\eta$ ratios, ranging from 1:3 to 3:1, were considered.

$$FM = -\Delta\eta\sqrt{1+(\Delta\Gamma:\Delta\eta)^2} \tag{2}$$

The 127080 dataset was divided in to three groups: 70% for training, 15% for validation and 15% for test. For further testing the generalization performance of the estimators, a different set was generated from bleed valve leakage faults. Up to 7% leakage faults were considered for each compressor to generate 140 sample points in total. This dataset is referred in this paper as a blind test case data.

3. RESULTS AND DISCUSSION

3.1. For the ARIMA model

The first part of the proposed method is faulty signal reconstruction using ARIMA. This part aims to correct sensor fault/failure corrupted measurements via the concept of time-series forecasting. When a sensor malfunction is detected, the measurements of that sensor before the malfunction is started are, in principle, considered and the actual values for the affected timesteps are forecasted. However, since a delay in the detection is inevitable, typically for drift fault scenarios, some previous measurements should be discarded. This is because the future timestep predictions are influenced primarily by the current and the closest previous measurements in the time-series.

Two different scenarios were taken into consideration. The first is when a sensor fault/failure occurs while the engine is undergoing through the state of gradual deterioration. The second is when a sensor fault/failure occurs together with a component fault. Gradual deterioration is a natural phenomenon that a gas turbine engine undergoes over its lifetime. This causes slow and simultaneous changes on the gas path measurements with time. Measurement changes are used to track performance trends, but at the same time, a sensor can also fail or become faulty. To be able to track the performance trend of the engine with no interruption, the corrupted data should automatically be corrected.

During training, three different datasets were considered. The first set was used to train the ARIMA model. Unseen dataset was then used to evaluate the performance of the trained model. Once the validation step was completed, the model was used to forecast future timestep values. Figure 4 shows the results obtained for the IPC delivery temperature in

comparison with the original dataset. The x-axis represents timesteps in “flight cycles” and the y-axis the measured values. The first 1500 data points (red line with diamond marks) indicate the training results, the next 500 points (green line with asterisk) show the test results, the following 500 points (blue line with plus sign marks) refer to the forecasted values, and the black label dedicated to the original dataset with no sensor fault/failure effects. It can be seen in Figure 4 that the forecasted values are more accurate and smoother than the measured values. The results from the remaining sensors are not included here due to space limitation.

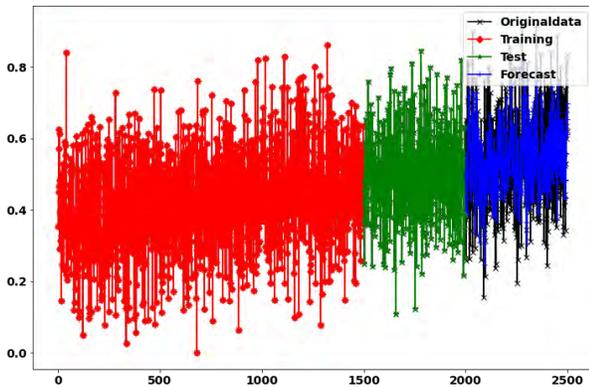


Figure 4. T25 fault/failure correction through forecasting using ARIMA.

In the second problem scenario, a sensor fault/failure occurrence in a faulty engine was analysed. When a component fault occurs, a set of measurements show considerable deviations from the gradual trend. The evolution period of component faults is shorter than the gradual deterioration. If one of the important sensors is failed or become faulty simultaneously with a component fault (although less likely to occur), the ARIMA model will correct the sensor problem. As illustrated in Figure 5, regardless of knowing the type of the sensor malfunction (bias, drift, or complete failure), just based on the detection information, the ARIMA model predicts the actual measurement using the faulty engine data collected before the sensor malfunction is detected. Avoiding the concern to identify the type and magnitude of the sensor malfunction reduces the complexity of the diagnostic problem and computational time. Figure 6 shows training results (red line with diamond marks), test results (green line with asterisk marks) and forecasting results (blue line with plus sign marks) obtained for N4 compared to the original dataset. Similar performances were recorded for the remaining sensors as well, but they are not presented in this paper due to space limitations. The ARIMA model was able to forecast N4 for the 100 future values with 2.13 RMSE. If required, it can also continue forecasting the values after the 100 future flight cycles without losing its accuracy. Moreover, it can be seen in Figure 6 that the forecasted values are smoother than the measured values due to the MA model.

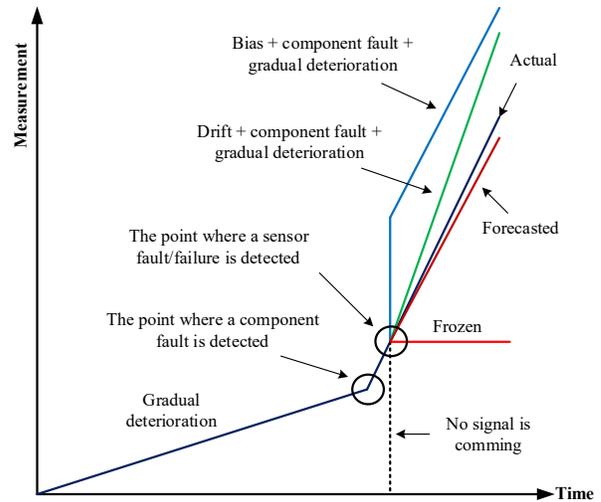


Figure 5. Schematic illustration of a sensor malfunction in a deteriorating faulty engine.

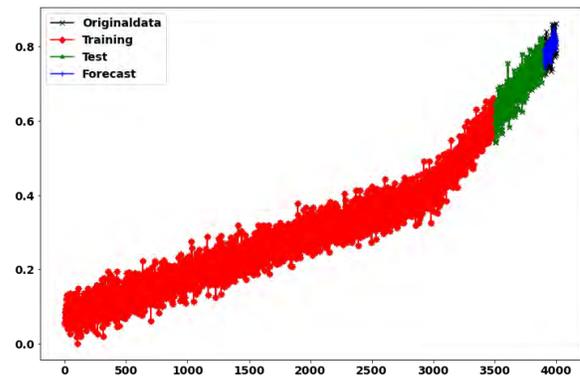


Figure 6. N4 fault/failure correction when a malfunction is detected on N4 in a deteriorating faulty engine.

3.2. For the NN models

Four individual NN models are developed to replace T23, P23, P43, and P46. These four sensors are considered physically unavailable for the turbofan engine. However, they are among the key measurements to distinguish between the Fan, IPC, HPC, HPT, IPT, and LPT faults and accurately estimate their severity. Two different datasets were used to train and verify the NN models. First, the 127080 dataset was randomly divided into three groups: 70% for training, 15% for cross-validation and the remaining 15% for test. The estimation accuracy of the NN models were examined based on the MAE and σ of the estimation errors in percent. These two parameters are selected to compare the prediction error with that of the Gaussian noise imposed to the data. Figure 7 shows 100 sample normalized test prediction errors. It compares the estimation errors with the measurement noise incorporated. It is seen that the predicted values are smoother and less affected by noise than the measured ones. This is also shown in Table 2 that all estimation error σ values are lower than the associated measurement noise values considered.

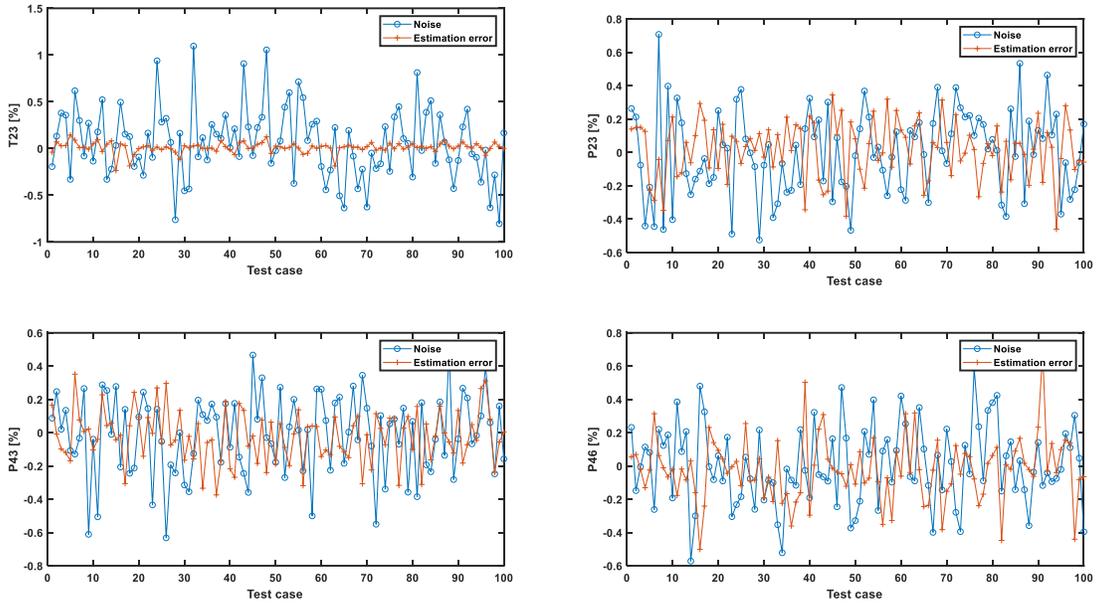


Figure 7. Estimation errors vs. the measurement noise incorporated, 100 random test cases out of the 19,062 test cases.

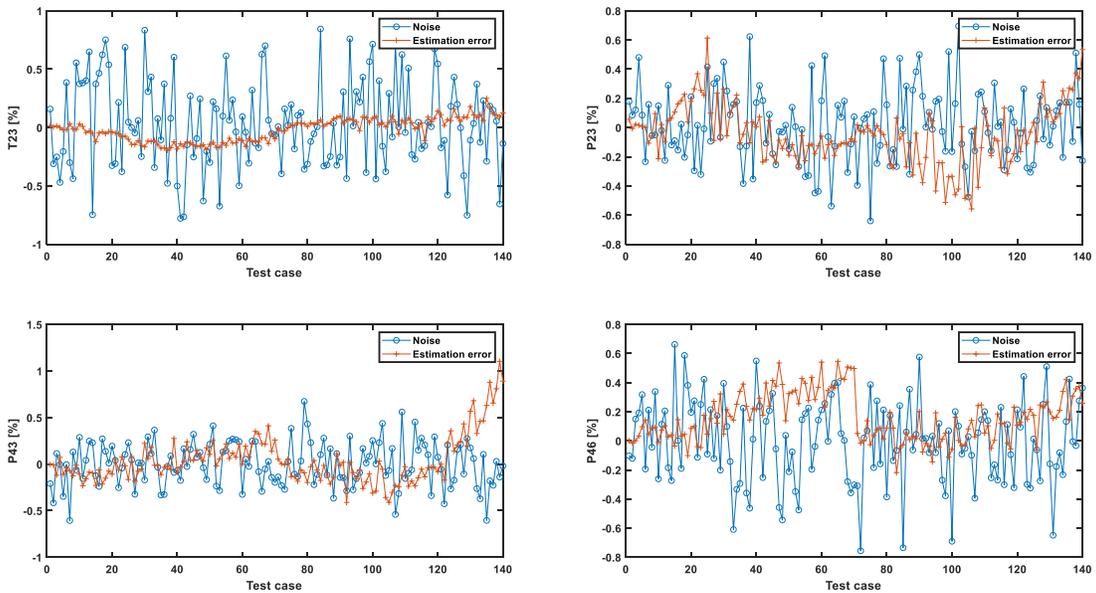


Figure 8. Estimation errors vs. the measurement noise incorporated, for the leakage fault data set.

Table 2. Prediction error vs. measurement noise added.

Error	T23	P23	P43	P46	Reference
Prediction MAE	0.006	-0.002	0.000	-0.001	Measured value
	-0.002	0.003	0.001	0.004	Actual value
Prediction error σ (%)	0.399	0.315	0.297	0.294	Measured value
	0.043	0.189	0.167	0.157	Actual value
Noise σ (%)	0.4	0.25	0.25	0.25	

When applying the blind test data, the results shown in Figure 8 were obtained. The first 70 points represent the estimated measurements associated with the IPC leakage fault and the cases from 71 to 140 are the estimated measurements for the HPC leakage data. To show how far the measurements with leakage faults are from the 127080 training data feature space, the equivalent performance parameter deviations for the Fan, IPC, HPC, HPT, IPT, and LPT were estimated through an adaptive scheme and presented in Figure 9. The equivalent fault magnitude estimated for the Fan, HPT, and IPT reaches up to 8.5%, which is greater than twice of the

fault magnitude considered for the 127080 data. For the HPC, it is approximately 1.6 times and for the IPC and the Fan 1.2 times the fault magnitude used to generate the 127080 set. This proves the generalization capability of the NN estimators developed. Enlarging the training domain could also further improve the estimation accuracy of the models.

The effect of missing one sensor among T25, P25, T3, P3, T46 and T5 on the accuracy of T23, P23, P43, and P46 estimators was also analysed. Figure 10 and Table 3 show test results for the leakage fault data. For all the measurements, estimation errors often fall within the threshold of the measurement noise incorporated. As shown in Figure 10, the error increases with increasing the magnitude of the leakage faults. Particularly, when the effect of the leakage faults on the input measurements excides the equivalent component fault effects. This is expected since the input data distribution starts significantly shifting from the training data space. Although the obtained accuracy is satisfactory, it could also be further improved by increasing the training feature space.

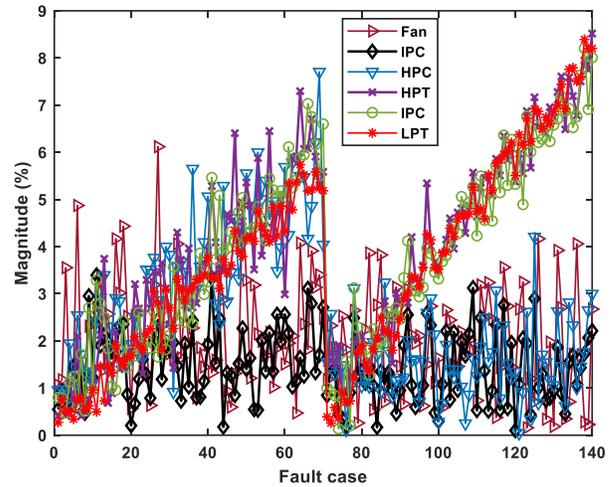


Figure 9. Performance parameter deviations induced by the IPC and HPC leakage faults.

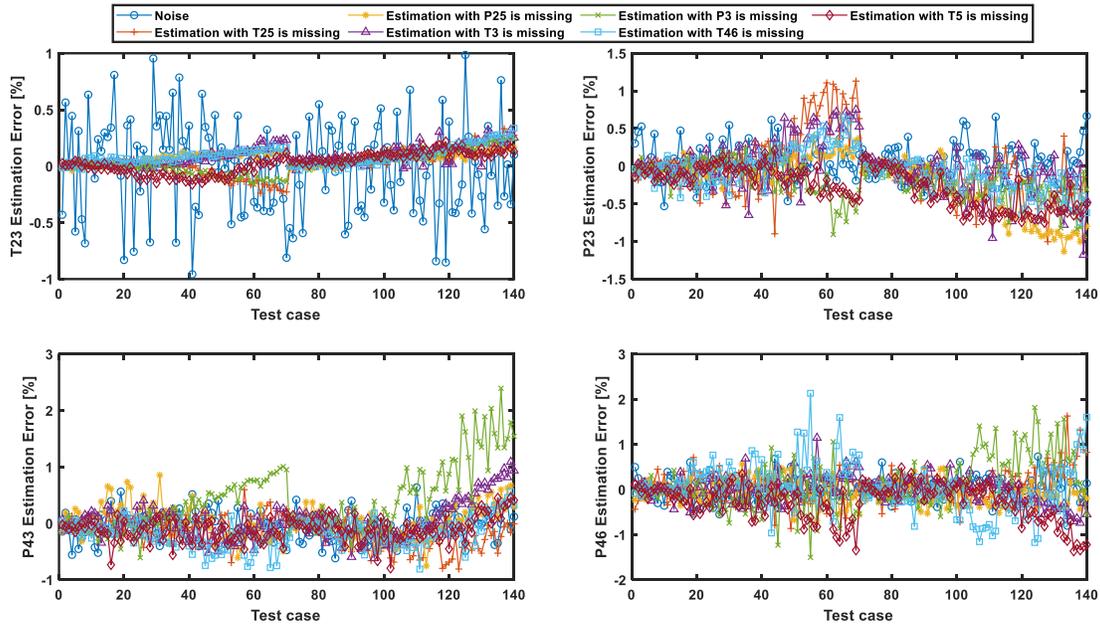


Figure 10. T23, P23, P43, P46 test results for the leakage dataset when one of T25, P25, T3, P3, T46, and T5 is missing.

Table 3. Calculated RMSE values for estimated T23, P23, P43, and P46 measurements when one of the T25, P25, T3, P3, T46 and T5 is missing, for the leakage fault dataset.

Estimated	Missing sensor					
	T25	P25	T3	P3	T46	T5
T23	0.143	0.442	0.594	0.275	0.489	0.441
P23	0.253	0.251	0.398	0.226	0.194	0.427
P43	1.809	2.952	1.332	2.630	1.480	1.941
P46	0.550	1.718	0.581	2.567	1.299	1.506

The capability of the NN models to accurately estimate T25, P25, T3, P3, T46 and T5 when one among them is missing was also investigated. If, for example, T25 is missing, it will be estimated from the remaining measurements: N4, N44, N46, P25, T3, P3, T46, and T5. The standard deviation of the estimation error for each of the six NN modules, for the leakage fault dataset, is presented in Table 4. It is seen in that the estimation error for T25, T46 and T3 is approximately half times the measurement noise considered. The estimation error for T5, on the other hand, is higher, almost twice of the measurement noise incorporated. This is expected because

T5 is shown to have very low correlation with T25, P25, T3, and P3. Similarly, the estimation error for P25 and P3 is higher. The reason could be since P25 and P3 are the most useful measurements to estimate bleed leakages and flow capacity in the two compressors, and since they are also the only pressure measurements available on the gas path, estimating them based only on the other temperature and speed measurements seems relatively difficult. In particular, the estimation error associated with the IPC bleed leakage data was higher than the HPC bleed leakage data for both P25 and P3. For the T25 case, the error for the IPC leakage data was equivalent to a soft sensor fault. Whereas for the HPC leakage data, the error was close to the measurement noise added. Including some leakage induced measurements in the training data sample may improve the accuracy of the two pressure estimators.

Table 4. T25, P25, T3, P3, T46 and T5 estimation accuracy for the leakage fault data set vs. measurement noise added.

Error σ (%)	Estimated measurement					
	T25	P25	T3	P3	T46	T5
Estimation	0.245	1.000	0.199	0.983	0.256	0.777
Noise	0.400	0.250	0.400	0.250	0.400	0.400

4. CONCLUSION

A sensor fault/failure correction and missing sensor replacement method is proposed for three-shaft turbofan engines. In this method, Autoregressive integrated moving average and feedforward neural networks are utilized. The former is responsible to correct faulty measurements through time-series forecasting. A set of neural network models are also devised to replace important sensors for diagnostics which are not installed from the beginning or missed through time due to maintenance activities and damages. When the engine is brand new equipped with full instrumentation suite, the neural network models are used to replace additional measurements required for more advanced diagnostic solutions. Another set of neural network models is developed to replace missing sensors sometime in the engine life cycle. This kind of sensor fault/failure correction and missing sensor replacement system is important in real-time to enable a continuous engine monitoring and diagnostic process with no interruption for re-calibration and re-installation. Additionally, the missing measurement may result in underdetermined problem, indistinguishable failure modes, and inaccurate severity level estimation results.

Performance data generated from the turbofan engine model under different degradation scenarios were used to train and evaluate the method proposed. For all scenarios considered, the faulty sensor signals were able to be corrected successfully with reconstruction errors lower than the measurement noise incorporated. Similarly, for most of the neural network estimators developed, the standard deviation of the test errors was lower than the measurement noise

standard deviation considered. The method proposed is flexible for modification to accommodate different engine configurations. However, future work should assess the impact of the sensor scheme on the engine gas path analysis. Ambient condition and operating condition variations were not also included in this paper. Moreover, a linear gradual deterioration profile was considered while gas turbines exhibit nonlinear behavior.

ACKNOWLEDGEMENT

This research was funded by the Swedish Knowledge Foundation (KKS) under the project PROGNOSIS, Grant Number 20190994.

REFERENCES

- Afman, J.-P., Prasad, J. V. R., & Antolovich, S. (2017). Real-Time Virtual Sensing of Component Damage Variables in a Gas Turbine Engine. *ASME 2017 Gas Turbine India Conference*, December 7–8, Bangalore, India, V002T11A002. Doi: 10.1115/GTINDIA2017-4706.
- Aivaliotis, P., Georgoulas, K., Arkouli, Z., & Makris, S. (2019). Methodology for enabling digital twin using advanced physics-based modelling in predictive maintenance. *Procedia Cirp*, 81, 417-422. Doi: 10.1016/j.procir.2019.03.072.
- Bettocchi, R., Pinelli, M., Spina, P. R., & Venturini, M. (2006). Artificial Intelligence for the Diagnostics of Gas Turbines—Part I: Neural Network Approach. *Journal of Engineering for Gas Turbines and Power*, 129(3), 711-719. Doi:10.1115/1.2431391.
- Fentaye, A., Zaccaria, V., & Kyprianidis, K. (2021). Discrimination of rapid and gradual deterioration for an enhanced gas turbine life-cycle monitoring and diagnostics. *International Journal of Prognostics and Health Management*, 12(3). Doi: 10.36001/ijphm.2021.v12i3.2962.
- Fentaye, A., Zaccaria, V., Rahman, M., Stenfelt, M., & Kyprianidis, K. (2020). Hybrid Model-Based and Data-Driven Diagnostic Algorithm for Gas Turbine Engines. *Proceedings of ASME Turbo Expo 2020 Turbomachinery Technical Conference and Exposition*, September 21–25, 2020, GT2020-14481. Doi: 10.1115/GT2020-14481.
- Fentaye, A. D., Baheta, A. T., Gilani, S. I., & Kyprianidis, K. G. (2019). A Review on Gas Turbine Gas-Path Diagnostics: State-of-the-Art Methods, Challenges and Opportunities. *Aerospace*, 6(7), 83. Doi: 10.3390/aerospace6070083.
- Ganguli, R. (2002). Fuzzy logic intelligent system for gas turbine module and system fault isolation. *Journal of propulsion and power*, 18(2), 440-447. Doi: 10.2514/2.5953.
- Jasmani, M. S., Li, Y.-G., & Ariffin, Z. (2011). Measurement Selections for Multicomponent Gas Path Diagnostics Using Analytical Approach and Measurement Subset

- Concept. *Journal of Engineering for Gas Turbines and Power*, 133(11). Doi:10.1115/1.4002348.
- Jiang, Y., Yin, S., Dong, J., & Kaynak, O. (2021). A Review on Soft Sensors for Monitoring, Control, and Optimization of Industrial Processes. *IEEE Sensors Journal*, 21(11), 12868-12881. Doi:10.1109/JSEN.2020.3033153.
- Jombo, G., Zhang, Y., Griffiths, J. D., & Latimer, T. (2018). Automated Gas Turbine Sensor Fault Diagnostics. *Proceedings of ASME Turbo Expo 2018: Turbomachinery Technical Conference and Exposition*, June 11–15, 2018, Oslo, Norway, GT2018-75229, V006T05A003. Doi: 10.1115/GT2018-75229.
- Kaboukos, P., Oikonomou, P., Stamatis, A., & Mathioudakis, K. (2003). Optimizing diagnostic effectiveness of mixed turbofans by means of adaptive modelling and choice of appropriate monitoring parameters. *RTO A VT Symposium on "Ageing Mechanisms and Control: Part B - Monitoring and Management of Gas Turbine Fleets for Extended Lift and Reduced Costs"*, Manchester, UK, 8-11 October 2001, and published in RTO-MP-079(1).
- Kamat, S., & Madhavan, K. P. (2016). Developing ANN based Virtual/Soft Sensors for Industrial Problems. *IFAC-PapersOnLine*, 49(1), 100-105. Doi:https://doi.org/10.1016/j.ifacol.2016.03.036.
- Kramer, M. A. (1992). Autoassociative neural networks. *Computers & Chemical Engineering*, 16(4), 313-328. Doi:https://doi.org/10.1016/0098-1354(92)80051-A.
- Kyprianidis, K. (2017). An Approach to Multi-Disciplinary Aero Engine Conceptual Design. Paper presented at the *International Symposium on Air Breathing Engines*, ISABE 2017, Manchester, United Kingdom, 3-8 September 2017 Paper No. ISABE-2017-22661.
- Li, J., & Ying, Y. (2020). Gas turbine gas path diagnosis under transient operating conditions: A steady state performance model based local optimization approach. *Applied Thermal Engineering*, 170, 115025. Doi:https://doi.org/10.1016/j.applthermaleng.2020.115025.
- Li, Y. G. (2002). Performance-analysis-based gas turbine diagnostics: A review. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, 216(5), 363-377. Doi:10.1243/095765002320877856.
- Lu, F., Li, Z., Huang, J., & Jia, M. (2020). Hybrid State Estimation for Aircraft Engine Anomaly Detection and Fault Accommodation. *AIAA Journal*, 58(4), 1748-1762. Doi: 10.2514/1.J059044.
- Marinai, L. (2004). *Gas-path diagnostics and prognostics for aero-engines using fuzzy logic and time series analysis*. Ph.D. Thesis, Cranfield University, Bedford, UK. <http://dspace.lib.cranfield.ac.uk/handle/1826/6730>.
- Marinai, L., Singh, R., Curnock, B., & Probert, D. (2003). Detection and Prediction of the Performance Deterioration of a Turbofan Engine. In: *TS-005, International Gas-turbine Congress 2003 Tokyo*; November 2–7, 2003.
- Ogaji, S. O., & Singh, R. (2003). Advanced engine diagnostics using artificial neural networks. *Applied soft computing*, 3(3), 259-271. Doi: 10.1016/S1568-4946(03)00038-3
- Sadough Vanini, Z. N., Meskin, N., & Khorasani, K. (2014). Multiple-Model Sensor and Components Fault Diagnosis in Gas Turbine Engines Using Autoassociative Neural Networks. *Journal of Engineering for Gas Turbines and Power*, 136(9). Doi:10.1115/1.4027215.
- Saias, C. A., Pellegrini, A., Brown, S., & Pachidis, V. (2021). Three-spool turbofan pass-off test data analysis using an optimization-based diagnostic technique. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, 235(6), 1577-1591. Doi:10.1177/09576509211002311.
- Simon, D. L., & Garg, S. (2009). Optimal Tuner Selection for Kalman Filter-Based Aircraft Engine Performance Estimation. *Journal of Engineering for Gas Turbines and Power*, 132(3). Doi:10.1115/1.3157096.
- Simon, D. L., & Rinehart, A. W. (2016). Sensor Selection for Aircraft Engine Performance Estimation and Gas Path Fault Diagnostics. *Journal of Engineering for Gas Turbines and Power*, 138(7). Doi:10.1115/1.4032339.
- Stenfelt, M., & Kyprianidis, K. (2022). Estimation and Mitigation of Unknown Airplane Installation Effects on GPA Diagnostics. *Machines*, 10(1), 36. Retrieved from <https://www.mdpi.com/2075-1702/10/1/36>.
- Zaccaria, V., Fentaye, A. D., Stenfelt, M., & Kyprianidis, K. G. (2020). Probabilistic model for aero-engines fleet condition monitoring. *Aerospace*, 7(6), 66. Doi: 10.3390/aerospace7060066.
- Zhou, J., Liu, Y., & Zhang, T. (2016). Analytical Redundancy Design for Aeroengine Sensor Fault Diagnostics Based on SROS-ELM. *Mathematical Problems in Engineering*, 2016, 8153282. doi:10.1155/2016/8153282.
- Zhou, X., Lu, F., & Huang, J. (2019). Fault diagnosis based on measurement reconstruction of HPT exit pressure for turbofan engine. *Chinese Journal of Aeronautics*, 32(5), 1156-1170. Doi:https://doi.org/10.1016/j.cja.2019.03.032.

Helicopter Bolt Loosening Monitoring using Vibrations and Machine Learning

Eli Gildish¹, Michael Grebshtein², Yehudit Aperstein³, Alex Kushnirski⁴, and Igor Makienko⁵

^{1,2,5}*RSL Electronics LTD, Migdal Ha'Emek, 23100, Israel*

elig@rsl-electronics.com
michaelg@rsl-electronics.com
igor@rsl-electronics.com

³*Afeke Tel-Aviv College of Engineering, Tel Aviv-Yafo, 6910717, Israel*

apersteiny@afeke.ac.il

⁴*Israeli Air Force, Tel Aviv-Yafo, IAF HQ, Israel*

alekush@idf.gov.il

ABSTRACT

The existing helicopter Health and Usage Management Systems (HUMS) collect and process flight operational parameters and sensors data such as vibrations to provide health monitoring of the helicopter dynamic assemblies and engines. So far, structure-related mechanical faults, such as looseness in bolted structures, have not been addressed by vibration-based condition monitoring in existing HUMS systems. Bolt loosening was identified as a potential risk to flight safety demanding periodical visual monitoring, and increased maintenance and repair expenses. Its automatic identification in helicopters by using vibration measurements is challenging due to the limited number of known events and the presence of high-energy vibrations originating in rotating parts, which shadow the low-level signals generated by the bolt loosening.

New developed bolt loosening monitoring approach was tested on HUMS vibrations data recorded from the IAF AH-64 Apache helicopters fleet. ML-based unsupervised anomaly detection was utilized in order to address the limited number of faulty cases. The predictive power of health features was significantly improved by applying the Harmonic filtering differentiating between the high-energy vibrations generated by rotating parts compared with the low-energy structural vibrations. Different unsupervised anomaly detection techniques were examined on the dataset. The experimental results demonstrate that the developed approach enable successful bolt loosening monitoring in

helicopters and can potentially be used in other health monitoring applications.

1. INTRODUCTION

Health and Usage Management Systems (HUMS) in helicopters use permanently installed vibration sensors to perform continuous health monitoring and predict failures in dynamic assemblies. Unfortunately, mechanical looseness monitoring is not in scope by vibration monitoring in helicopters (CAP 753, 2018).

IAF's field experience shows that not all bolts in helicopters have secure wiring and existing monitoring solutions like visual inspections including torque checks are not cost effective, performed on ground only and are subject for human errors. The use of vibration sensors appears to be the most cost-effective solution among all the alternatives (including additional sensors installation) given the existence of HUMS vibration sensors kit on the helicopter.

1.1. Mechanical Looseness

The literature divides mechanical looseness into three types (VibrAlign, 2019): A, B and C where each type is characterized by different changes in vibrations spectrum. The spectrum Type A mechanical looseness manifests itself as an increase in the amplitude of shaft's first harmonic (1X), while Type B and C affect the energy of shaft harmonics and subharmonics (e.g. 0.5X) often characterized by a raised vibrations noise floor as a result of changes in system natural frequencies.

Jackson (1996) and Human (2011) suggested that looseness is not a simple phenomenon and undergo certain stages, as the condition of the equipment deteriorates. Initially

Eli Gildish et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

looseness will manifest itself as Type A, next as B, then as C, and finally only the noise floor will remain in the spectrum. Krot et al. (2020) and He et al. (2014) also found that bolt loosening is best diagnosed by using system natural frequencies.

1.2. Looseness Isolation

Bolt loosening detection by using vibration monitoring has been investigated in previous studies and ML methods were successfully applied (Eraliev et al., 2022). Unfortunately, the separation capability between bolt loosening and other mechanical failures hasn't been addressed. A reliable bolt loosening isolation is required by helicopter operators to reduce false alarms and improve helicopter maintenance.

As mentioned in 1.1 the changes in vibration noise floor, related to system natural frequencies, play a major role in bolt loosening detection. In this study, the vibration noise floor will be isolated from the periodic vibrations related to rotating parts to allow separation between looseness and other mechanical failures.

The methods below enable the separation between vibration background noise and periodical vibration signals. Antoni et al. (2004) and Randall (2004) exploited the periodic nature of the signal in order to build an adaptive filter that rejects uncorrelated noise between time-shifted slices of the signal. Such a filter quickly becomes impractical as it naturally grows with signal's complexity. Another method pioneered by Randall et al. (2011) is cepstrum-based separation of discrete components. Cepstrum is the spectrum of log spectrum. Transform to the cepstrum domain moves all the harmonics of the same shaft to a known place in queffency (the frequency analog of cepstrum), where these harmonics can be easily removed. Although the method is generic, it is not very accurate, and requires a large number of harmonics per shaft to appear in the spectrum. Groover et al. (2005), Braun (2011), and Peeters et al. (2005 and 2007) remove periodic content by repeatedly resampling the signal to a constant angular basis, removing the bin-centered peaks in the order domain and resampling back to the constant time domain. Groover's method is accurate and better suited for signals with a large number of periodic sources like vibrations, but requires knowledge about system kinematics.

1.3. ML-based Anomaly Detection

In aviation applications, plenty of normal recordings exist with only a small number of abnormal events. Thus, it would be correct to define the bolt loosening detection as an anomaly detection problem. The anomaly detection methods learn the normal data behavior and identify points, sequences or context that deviate from the normal behavior (Goldstein et al., 2016 and Barelli et al., 2021).

In recent years, there has been a rapid growth in application of anomaly detection techniques in aviation (Basora et al.,

2019 and Basora et al., 2021) where the unsupervised learning techniques are widely used in analysis of mechanical vibration data. Xu et al. (2019) proposed anomaly detection method in vibration signals collected from a certain type of rolling bearing equipment. Camerini et al. (2018) developed one-class classification SVDD for detection of micro-pitting damage on a helicopter gear. Lee, G. et al. (2020) developed unsupervised anomaly detection method for diagnosis of industrial gas turbines. The authors included in their model different types of data collected from vibration transmitters, temperature and pressure transmitters. Unsupervised anomaly detection models developed by Park et al. (2019) for vibration diagnostics of washing machine and by Oliveira et al. (2019) using 257 attributes, such as real measurements from thermal, acoustic and impact sensors installed in a heavy haul railway line in Brazil. Principi et al. (2019) presented unsupervised method for diagnosing faults of electric motors. Hu et al. (2020) proposed the features extraction method for fault detection based vibration signals of rotating machinery where the features obtained by vibrations FFT from classic rotor and bearing datasets are used as inputs to KCPA and AE models.

There is a wide use of reconstruction-based methods in mechanical fault diagnosis. Liu et al. (2018) and Sun et al. (2019) implemented this approach for the rolling bearing diagnosis and Ma et al. (2020) for damage identification task of a bridge under moving vehicle.

2. PURPOSE AND PROBLEM DEFINITION

The purposes of the current study are as follows:

1. Developing a new methodology for bolt-loosening detection by using vibration sensors already existing in helicopters
2. Enabling reliable bolt loosening isolation from other mechanical problems
3. Testing the methodology by using field data recorded by HUMS

The following challenges were addressed in this research:

1. The helicopter maintenance information in digital form was difficult to access limiting thus the labeling of normal observations.
2. The missing information about the start time of the bolt loosening events makes the abnormal data labeling to be difficult
3. The number of abnormal observations is usually limited in helicopter operations since maintenance actions are scheduled to prevent mechanical failures. As a result, the supervised ML methods requiring significant statistics of abnormal observations cannot be used.
4. Bolt loosening isolation requires separation between vibration noise and shaft-synchronized vibration components (see Section 1.2). However, the background

noise and faulty bearing vibrations cannot be separated since the last are non-synchronized to shaft speeds. On the other hand, given a lack of comprehensive maintenance information, the faulty bearings may be a part of the observations where bolt loosening does not exist.

3. METHODOLOGY

The flowchart in Figure 1 describes the proposed methodology. The feature extraction algorithm generates two datasets (raw and advanced) of the same size: with and without Harmonic filtering to evaluate its influence on the model accuracy. The datasets are divided into training and testing sets where healthy data is divided 50/50 between the sets and the faulty data was fully a part of the test set to enable model performance evaluation. Three unsupervised ML models: Naïve model (less accurate and used as a baseline for models performance evaluation), Isolation Forest and Auto Encoder were built by using the train data and evaluated on the test data. The model evaluation was performed by using AUC criterion given the predicted and expected data labels.

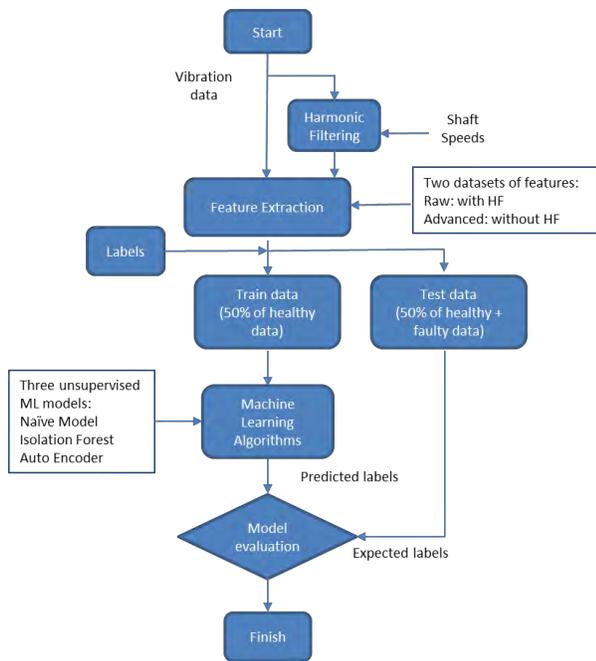


Figure 1. Flowchart demonstrating a proposed methodology

3.1. Harmonic Filtering

The Harmonic filtering (HF) (Groover at al., 2005) was chosen as the best preprocessing improvement to the feature extraction since it provides the best accuracy given the information about system shaft speeds. HF removes the periodic vibration components related to the rotating parts while leaving the background noise related to system natural frequencies (as mentioned in Section 1.2).

The technique consists of the following steps and repeated for each shaft in helicopter drive-train:

1. Times that correspond to the constant angular intervals $\Delta\theta$ are determined from the angular speed of rotating shaft
2. Vibration signal, sampled at constant time intervals Δt , is interpolated into a constant angular basis (cycle domain)
3. The FFT is applied to the interpolated signal
4. All the shaft harmonics, which are now exactly bin centered, are removed from the spectrum
5. The cleaned signal is transformed back to cycle domain via inverse FFT
6. Vibration signal is interpolated back to the time domain, where time intervals Δt are constant

3.2. Order Tracking

Order tracking, as used in the feature extraction, is a method of vibration analysis. The spectral density is calculated in terms of shaft speed (orders) instead of frequency (Hz). Order tracking helps to identify speed-related vibrations such as shaft, gearwheel and bearing defects. The order tracking requires vibration signal to be converted into cycle domain instead of time domain where signal is sampled at constant increments of shaft angle instead of constant increments of time (as described in Section 3.1) and then the spectral density is calculated by using Power Spectral Density (PSD) estimation. More information about the method can be found in Fyfe et al. (1997).

3.3. Feature Extraction

Two different features datasets: raw (without HF) and advanced (by using HF prior to feature extraction) were calculated.

The features extraction consisted of the following steps:

- The order tracking was performed (see Section 3.2)
- The order domain was limited between 0 and 160Hz to eliminate the influence of bearing fault frequencies, which expected to appear at higher frequency band (see Section 2) and divided into M equally spaced bands.
- The Root Mean Square (RMS) was calculated for each band in the order domain resulting in M features per sensor. The RMS at band i was calculated as follows:

$$RMS_i = \sqrt{\frac{1}{N} \sum_{j=1}^N a(j)}, i \leq M \quad (1)$$

where RMS_i is RMS of band i , N is a number of bins in band i , and $a(j)$ is a vibrations PSD at bin j of the order domain.

$M=20$ was chosen to achieve a trade-off between the model flexibility (requiring high M) and model complexity (requiring low M). Each dataset dimension is equal to 60 (3sensors x 20features).

3.4. Data Labeling

The following solutions of data labeling are proposed to solve the challenges as described in Section 2:

1. Assuming the significant majority of healthy observations in data, all the observations except the known bolt loosening events were labeled as “normal”. However, the accuracy of the False Positive (FP) rate estimation is expected to be limited since other mechanical failures may generate FP alerts.
2. Given lack of information about bolt loosening start, the abnormal data period was labeled roughly by using a visual inspection. Thus, the estimation of model True Positive (TP) rate is limited when using this type of labeling.

3.5. ML Approach

The use of anomaly detection techniques is chosen as a best way to detect the problems since the abnormal data amount is limited as mentioned in Section 2. The anomaly detection allows learning from healthy observations only and detecting abnormal observations as anomalies.

Two advanced unsupervised ML techniques were chosen for comparison: Isolation Forest (IF) and Auto-Encoder (AE). The IF algorithm was developed specifically for the purpose of anomaly detection and works on the principle of isolating anomalies. AE allows learning a low-dimensional feature representation on which the given data instances can be well reconstructed. The reason for using AE in anomaly detection is that the learned feature representations are enforced to learn complex relations of the data to minimize reconstruction errors; anomalies are difficult to be reconstructed from the resulting representations and thus have large reconstruction errors. Both methods are widely used in conditional monitoring.

3.5.1. Isolation Forest (IF)

IF is based on an ensemble of random binary trees that compute paths to isolate observations. Each tree of the ensemble is known as an isolation tree, to partition observations until they are isolated. The identifying of a normal observation and abnormal observation by IF can be observed in Figure 2. A normal point (on the left) requires more partitions to be identified rather than an abnormal point (right). Therefore, the key idea is that anomalies are easier to isolate since they require shorter paths or fewer conditions in comparison with normal observations.

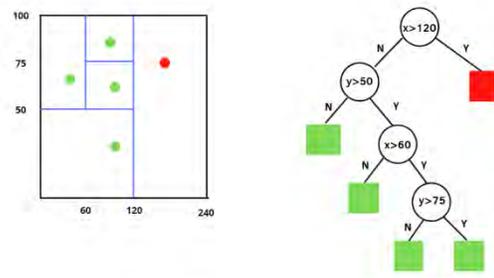


Figure 2. Identifying normal vs. abnormal observations with iForset

The algorithm determines limit values per node based on the feature values chosen randomly. At each step, the limits are split into 2 parts and being checked whether or not the sample observation is within the boundaries. IF determines anomalies scores by the following equation:

$$s(x, n) = 2 \frac{E(h(x))}{c(n)} \quad (2)$$

where $h(x)$ is the path length of observation x , $c(n)$ is the average path length of search in a Binary Search Tree and n is the number of external nodes. $E(h(x))$ is the mean value of $h(x)$ in the isolation tree. IF can be configured as a binary classifier or regression to weight the observation between normal (0) and anomalous (1). When configured as a classifier, it is possible to define the contamination proportion of outliers in the dataset used as a threshold to round the value to 0 or 1. Therefore, when s is close to 1, it is quite possible to be an outlier. Similarly, a number close to 0 might also be normal. If all observations are close to 0.5, no anomaly is identified (very similar observations).

More details about anomaly detection by IF can be found in Liu, F. T. et al. (2008).

3.5.2. Auto-Encoder (AE)

An auto-encoder is a type of neural network used to encode the data in efficient unsupervised manner. AE is trained to generate the target values equal to the input data through a combination of encoder and decoder networks. These networks have a bottleneck hidden layer of few neurons in the middle, forcing them to generate valid representations that compress the input data into a lower-dimensional code called a latent vector, which used by the decoder to reproduce the original information. AE is trained to minimize reconstruction errors

$$\min_{E,D} \|x - D(E(x))\| \quad (3)$$

where x is the input data, E is an encoder network, and D is a decoder network. The training of an auto-encoder is performed through backpropagation of the error, just like a regular feedforward neural network.

A typical auto-encoder architecture consists of three main components, as shown in Figure 3.

1. Encoder network: An encoder network is comprised of series of layers with a decreasing number of nodes and ultimately reduces input data into a latent vector. This process is also called dimensionality reduction, and the convolution layers are used for encoding.
2. Latent vector: The latent vector represents the lowest level space in which the inputs are reduced, with essential information preserved with the strong correlation between input features.
3. Decoder network: A decoder network acts as the mirror image of the encoder network. The number of nodes in every layer increases and reconstructs the latent vector to output as a similar input via transposed convolution. As a particular portion of the information is lost during reconstruction, the output data always have lower quality than the input data.

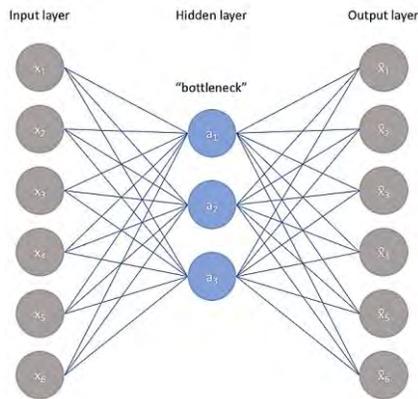


Figure 3. The typical AE configuration

The efficiency of AE in anomaly detection is strongly related to a proper selection of dimension size of hidden layers where anomalies are detected by calculating the residual error in the reconstruction of the input by the decoder. Since anomalies are “few and different”, AE tends to achieve lower error for normal observations and abnormal higher residuals for outliers. Similar to IF models, the contamination hyper parameter defines the percentage cut between normal and faults. The records with the highest residual errors are classified as abnormal. More detail about AE as an anomaly detection technique can be found in Chalapathy, R., & Chawla, S. et al. (2019).

3.6. Model Performance Evaluation

Technically, there is no way to measure the performance of unsupervised learning models, since there are no labels available to compare the ground truth. In this regard, we used abnormal signals as labels solely for model’s performance evaluation purposes. During the training process, labeled datasets are not provided to the models. The validation

dataset contained both normal and abnormal observations to use classification performance metrics.

Classification performance can be measured independently from threshold setting by introducing the receiver operating characteristic (ROC) curves. Such curves represent the fraction of target objects accepted by the model (i.e. normal observations classified as normal) against the fraction of outliers accepted (i.e. abnormal observations classified as normal). The area under the ROC curve (AUC) gives a scalar measure of the achieved separability between states.

4. DATASET AND EXPERIMENT DESCRIPTION

Intermediate Gearbox (IGB) represents one of the drive-train assemblies in AH-64 helicopters responsible for transfer of rotational moment from the main gearbox to the tail gearbox. The IGB is attached to the airframe by 4 bolts as shown in the Figure 4 and Figure 5.

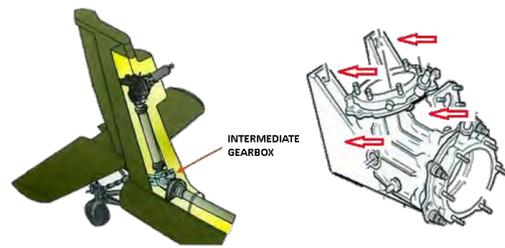


Figure 4. Intermediate Gearbox (IGB) of AH-64 and the corresponding locations of 4 bolts attaching the assembly to the airframe



Figure 5. Intermediate Gearbox of AH-64 and the upper left bolt example

Bolt loosening is manually checked before each flight as well as during weekly inspections. The vibrations levels of the IGB input shaft harmonics 1X and 2X are being monitored by HUMS continuously during flight accompanied by periodic manual measurements on-ground. Unfortunately, the existing IGB vibration monitoring does not provide a solution.

The bolt loosening events were detected indirectly by visual inspection of the bolt holes during IGB overalls. The oval form of the holes and signs of corrosion represent the result

of bolt loosening needed to be monitored automatically and in advance. See Figure 6 below for example.



Figure 6. The oval form detected in one of four holes points to bolt loosening

The historical data provided for this research was recorded by the T-HUMS of RSL Electronics Ltd. between 2014 and 2020 from the entire IAF AH64 helicopters fleet. Unfortunately, the data set cannot be published due to confidentiality limitations.

There are 3 vibration sensors in close proximity to IGB as summarized in Table 1.

Table 1. Vibration sensors in IGB proximity

Sensor Code	Sensor Description	Sensor Location	Samp Rate, kHz	Recording Duration
IGB	Intermediate Gearbox	IGB assembly, on IGB	48	10 sec
HBA	Aft Hanger Bearing	IGB input shaft, 0.5m from IGB	12	10 sec
HBF	Fwd Hanger Bearing	IGB input shaft, 1.5m from IGB	12	10 sec

There are four known cases of IGB bolt loosening in AH-64 where T-HUMS data exist, each with slightly different findings (see Table 2).

Table 2. IAF Findings of Bolt Loosening in IGB between 2014-2020

Case #	IGB Installation	IGB Removal	#Oval Holes Detected/ #Total Holes
1	10/07/18	19/12/19	4/4
2	13/11/18	22/10/19	3/4
No data	13/03/14	20/07/14	No data
3	25/12/17	06/05/18	3/4
4	26/03/19	09/03/20	0/4

The problem severity defined as a number of oval holes detected after IGB removal. For example, 4 oval holes in case #1 points to the higher problem severity vs. 0 holes as in case #4 where only low mounting torque was identified. Cases #2 and #3 are of similar severity with 3 oval holes each.

4.1. Data Preprocessing: HF

Figure 7 shows how the high-energy vibrations related to rotating parts are being removed with HF approach (Section 3.1).

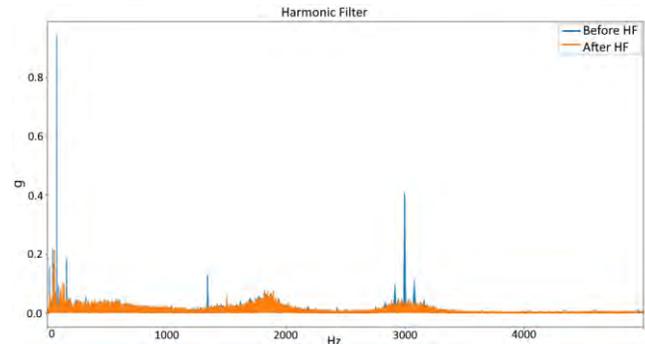


Figure 7. Example of IGB sensor vibrations spectrum before (blue) and after (orange) HF.

The noise floor difference between the normal and abnormal FFT of case#1 (see Table 2) is presented in Figure 8- where a minor noise floor increase can be visually identified.

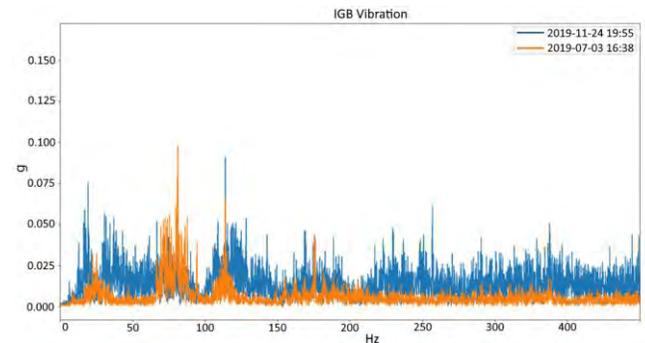


Figure 8. IGB sensor vibrations after applying HF: just after healthy IGB installation (orange) and before bolt loosening is detected (blue)

4.2. Data Statistics and Partition

The calculated features statistics after preprocessing is summarized in Table 3:

Table 3. Available Feature Dataset

Flight Regime	Description	Normal Observations	Abnormal Observations	Total Observations
1	Level Flight High Speed	44828	675	45503
2	Level Flight Low Speed	26371	371	26742
3	Hover	2714	39	2753

The only flight regime #1 with the largest number of observations (45,503 observations in total) was chosen to avoid influence of the helicopter operational conditions.

Further research is required to estimate the possibility of data aggregation from different flight regimes.

4.3. Benchmark Framework Definition

In order to perform ML model evaluation, the Naïve model was chosen as a baseline where other ML methods were evaluated by comparison to this baseline model. The model uses a probabilistic model assuming multivariate Gaussian statistics of the healthy data, estimates parameters of its probability distribution and assigns a log-likelihood to any observation during inference. The squared Mahalanobis distance is proportional to the log-likelihood and serves as an anomaly score for Naïve model, where large score values indicate a novelty in the data. To improve the model’s robustness, the score was calculated after the features are divided into 4 groups 5 features each assuming there is no dependency between the groups.

ROC curve (Receiver Operating Characteristic curve) was chosen to define the models performance and benchmarking. The ROC was calculated on the whole range of the model thresholds where True Positive Rate and False Positive Rate represented its axes. The ROC was built by using the test data including 50% of the healthy and all the faulty observations.

Area under the ROC Curve (AUC) was chosen to provide an indication of the model performance across all possible thresholds. Generally, a higher AUC points to a potentially better model performance. In our case, the area of interest is mainly in the region corresponding to a low FP rate (lower than 0.1) which is more practical for helicopter operators. The AUC of the Naïve model was equal to 0.8 as presented in Figure 6, where the model was built using all the 3 sensors and 20 features each (input dimension = 60).

Two different cases are presented in Figure 9: where the basic and advanced features datasets are used. The AUC in both cases for Naïve model is equal to 0.8. The AUC improvement is not significant while the low FP area was improved significantly by using HF pointing to its importance: 26% improvement in TP rate was found for a chosen 0.1 FP rate. Further model benchmarking results will be presented for HF only due to its significant improvement of the ROC for the low FP rate values.

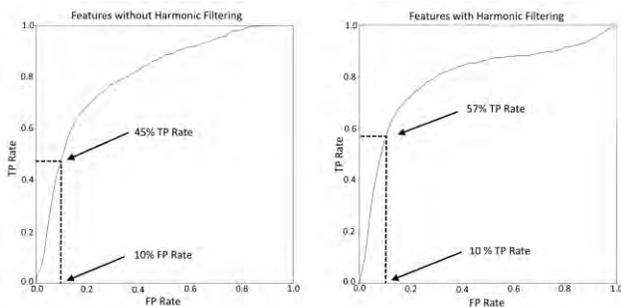


Figure 9. The ROC of Naïve model applied to basic features dataset (left) and advanced features dataset (right)

4.4. Advanced Models Settings

The performance of two advanced models: Isolation Forest and Auto-Encoder were compared to the baseline Naïve model.

The models were trained on the 50% of the healthy data (22,414 healthy observations) and the AUC was calculated by using the test data including other 22,414 healthy and 675 faulty observations.

IF algorithm, implemented on scikit learn library, used the default parameters except the contamination = 0.1 allowing the proportion of outliers in the dataset. Since its initialization is random, the AUC results were generated by averaging 100 runs.

Auto-Encoder algorithm was used with the default parameters except the contamination = 0.1 and hidden_neurons = [10, 4, 4, 10] where the parameter defines a number of neurons per hidden layer whose number is lower compared with the number of input features (60 in our case).

5. RESULTS SUMMARY

Table 4 below summarizes the results. As mentioned above, visual data investigation showed significant difference in the capability of the healthy-faulty data separation between different sensors. It was therefore decided to present the model benchmarking for each sensor separately. Table 4 summarizes the benchmarking and shows significant performance improvement when the IGB sensor only is used – 0.87 vs. 0.80 when all three sensors are used. All models performance on HBA and HBF sensors are poor compared to the IGB sensor. Further investigation is required to deeply understand the reason for the difference.

The stability of the model performance was estimated by randomly choosing subsets from the training set to generate AUC. Both Naïve and AE models were stable while IF model performance was different for different data sets. Given a similar performance between the models, the Naïve or AE model are preferred options for use in production. AUC values presented in Table 4 show that AUC for IF and AE models show no improvement vs. baseline Naïve model. The main reason for the models inaccuracy is the inaccuracy in dataset labeling due to lack of information about the bolt loosening beginning and info about mechanical failures in the healthy dataset. The best results are obtained when the IGB sensor only is being used.

Table 4. AI models benchmarking

Model	AUC – all sensors	AUC – HBA	AUC - HBF	AUC- IGB
Naïve model (Baseline)	0.80	0.61	0.66	0.86
Isolation Forest	0.79	0.51	0.64	0.87
Auto-Encoder	0.77	0.50	0.63	0.86

The IF ROC of IGB sensor is demonstrated in Figure 10. AUC of the IF model built on IGB sensor data is equal to maximal 0.87. The maximal TP rate of 0.75 can be obtained for a chosen 0.1 FP rate.

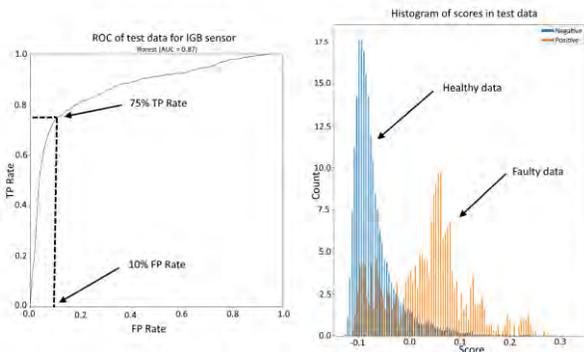


Figure 10. The best ROC of IGB sensor (left image) and histogram of anomaly scores (right image)

The difference in models performance between different sensors required additional data analysis. The histograms of the healthy and faulty data were analyzed in order to understand the difference in data statistics between the sensors. The difference in faulty and healthy data stats is more significant in IGB sensor. See example in one of the inputs in Figure 11.

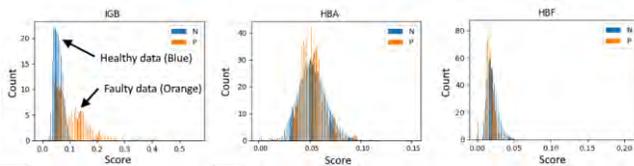


Figure 11. Comparison between the healthy and faulty data statistics for IGB (left), HBA (middle) and HBF (right) sensors

The best difference between the healthy and faulty data is visually recognized mainly in IGB sensor (Figure 11, left image) vs. HBA, HBF. The results fit also physical intuition since only IGB is located on the faulty assembly. As a comparison, the HBA and HBF data statistics are presented (Figure 11, center and right images).

The output of the models represents an anomaly score corresponding to a distance measure between each test observation and the training dataset.

The anomaly score can be used as a Condition Indicator in detecting bolt loosening. All the known bolt loosening cases were successfully detected by the Condition Indicator. Figure 12 shows an example of the Condition Indicator performance of AE model applied to all the data of the helicopter where the case #1 bolt loosening was found. The red line on the Figure 12 helps to identify the healthy and faulty data and is set to zero when no bolt loosening exists and equal to a high value in the period of IGB installation and removal when the problem appeared.

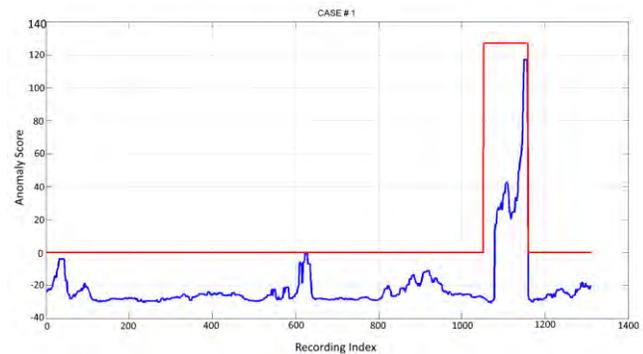


Figure 12. Example of Case #1 loosening detection by new Condition Indicator (blue)

6. CONCLUSIONS AND RECOMMENDATIONS

A new algorithm of bolt loosening detection by using vibrations and ML is developed and tested on field data recorded by IAF HUMS.

No significant difference in model performance was found between three ML models: Naïve model, Isolation Forest and Auto-Encoder. The models performance evaluation was limited due to dataset challenges as explained in Section 2.

The use of data from IGB sensor only outperforms the models using data from all three sensors. The actual reason is not clear yet and future research is required.

The use of HF as a preprocessing stage improves model TP rate especially for low FP rates where the model is of main interest for operators.

Recommendations for future research

The use of sequence-based anomaly detection instead of point-based may decrease FP rate where anomaly is detected if the samples neighborhood is considered.

Since data is time-based, the use of serial data structure may help to improve model performance. The use of time-based LSTM AE vs. AE as in our case may improve both FP and TP rates.

The use of more information about maintenance actions performed as well as the mechanical failures detected during the helicopters operations may improve significantly data labeling accuracy and improve models performance.

Further investigation of the difference in model performance between different sensors is required for deeper understanding the directions of model improvements.

Contribution

The new developed methodology allows bolt loosening detection and isolation from other mechanical failures. The methodology will improve helicopter maintenance activity by replacing non-reliable human factor used in periodical visual inspections and will expand monitoring scope also to flight conditions vs. existing manual ground-based inspections.

ACKNOWLEDGEMENT

The project has received funding from the DDR&D of Israel MOD under PHM innovation program.

REFERENCES

- Antoni, J., & Randall, R. (2004). Unsupervised noise cancellation for vibration signals: part II—a novel frequency-domain algorithm. *Mechanical Systems and Signal Processing*, vol. 18, no. 1, pp. 103-117.
- Barelli, E., & Ottaviani E. (2021). Unsupervised Anomaly Detection for Hard Drives. *Proceedings of the 6th European Conference of the Prognostics and Health Management Society*, pp. 10-16.
- Basora, L., Bry, P., Olive, X., & Freeman, F. (2021). Aircraft Fleet Health Monitoring with Anomaly Detection Techniques. *Aerospace*, 8(4), 103. <https://doi.org/10.3390/aerospace8040103>
- Basora, L., Olive, X., & Dubot, T. (2019). Recent advances in anomaly detection methods applied to aviation. *Aerospace*, 6(11), 117. <https://doi.org/10.3390/aerospace6110117>
- Braun, S. (2011). The Synchronous (Time Domain) Average revisited. *Mechanical Systems and Signal Processing*, vol 25(4), pp. 1087-1102.
- Camerini, V., Coppotelli, G., & Bendisch, S. (2018). Fault detection in operating helicopter drivetrain components based on support vector data description. *Aerospace Science and Technology*, 73, pp. 48-60. <https://doi.org/10.1016/j.ast.2017.11.043>
- CAP 753, (2018). Helicopter Vibration Health Monitoring. UK Civil Aviation Authority, Safety Regulation Group, ver. 2.
- Eraliev, O., Lee, K-H., & Lee, C-H. (2022). Vibration-Based Loosening Detection of a Multi-Bolt Structure Using Machine Learning Algorithms. *Sensors* 2022, 22, 1210, <https://doi.org/10.3390/s22031210>.
- Fyfe, K.R, and Munck, E.D.S. (1997). Analysis of computed order tracking. *Mechanical Systems and Signal Processing*, vol 11 no. 2, pp.187 – 205.
- Goldstein, M., & Uchida, S. (2016). A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. <https://doi.org/10.1371/journal.pone.0152173>
- Groover, C., Trethewey, M., Maynard, K. & Lebold, M. (2005). Removal of order domain content in rotating equipment signals by double resampling. *Mechanical Systems and Signal Processing*, vol. 19, no. 3, pp. 483-500.
- Jackson, K. (1996). Vibration Analysis Level 2 - Understanding the Basics. *Integrated Maintenance Solutions Inc.*
- Krot, P., Korennoi, V., & Zimroz, R. (2020). Vibration-Based Diagnostics of Radial Clearances and Bolts Loosening in the Bearing Supports of the Heavy-Duty Gearboxes. *Sensors* 2020, 20, 7284; [doi:10.3390/s20247284](https://doi.org/10.3390/s20247284).
- He, K., Zhu, W.D. (2014). Detecting loosening of bolted connections in a pipeline using changes in natural frequencies. *J. Vib. Acoust. Trans. ASME* 136, pp. 1–8.
- Human, E. (2011). *The Feasibility of Vibration Analysis as a Mechanism of Failure Analysis in Failure Investigation and Root Cause Analysis*. Magistral dissertation University of Johannesburg, Johannesburg, RSA.
- Hu, X., Xiao, Z., Liu, D., Tang, Y., Malik, O. P., & Xia, X. (2020). KPCA and AE Based Local-Global Feature Extraction Method for Vibration Signals of Rotating Machinery. *Mathematical Problems in Engineering*.
- Lee, G., Jung, M., Song, M., & Choo, J. (2020). Unsupervised anomaly detection of the gas turbine operation via convolutional auto-encoder. *In 2020 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pp. 1-6.
- Liu, H., Zhou, J., Xu, Y., Zheng, Y., Peng, X., & Jiang, W. (2018). Unsupervised fault diagnosis of rolling bearings using a deep neural network based on generative adversarial networks. *Neurocomputing*, 15, pp. 412-424.
- Ma, X., Lin, Y., Nie, Z., & Ma, H. (2020). Structural damage identification based on unsupervised feature-extraction via Variational Auto-encoder. *Measurement*, 160, 107811.
- Oliveira, D. F., Vismari, L. F., de Almeida, J. R., Cugnasca, P. S., Camargo, J. B., Marreto, E., ... & Neves, M. M. (2019). Evaluating unsupervised anomaly detection models to detect faults in heavy haul railway operations. *In 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 1016-1022.
- Park, S., Kang, J., Kim, J., Lee, S., & Sohn, M. (2019). Unsupervised and non-parametric learning-based anomaly detection system using vibration sensor data. *Multimedia Tools and Applications* 78, no. 4. Pp. 4417-4435.

- Peeters, B., Cornelis, B., Janssens, K. & Van der Auweraer H. (2007). Removing Disturbing Harmonics in Operational Modal Analysis. *Proceedings of the 2nd International Operational Modal Analysis Conference*. April 30 - May 2, Copenhagen, Denmark
- Peeters, B. & Van der Auweraer H. (2005). PolyMax: a revolution in operational modal analysis. *Proceedings of the 1st International Operational Modal Analysis Conference*. April 26-27, Copenhagen, Denmark.
- Principi, E., Rossetti, D., Squartini, S., & Piazza, F. (2019). Unsupervised electric motor fault detection by using deep autoencoders. *IEEE/CAA Journal of Automatica Sinica*, 6(2), pp. 441-451.
- Randall, R (2004). Unsupervised noise cancellation for vibration signals: Part I - Evaluation of adaptive algorithms. *Mechanical Systems and Signal Processing*, vol. 18, pp. 89-101.
- Randall, R., & Sawalhi, N. (2011). A New Method for Separating Discrete Components from a Signal. *Sound and Vibration*, vol. 45, pp. 6-9.
- Sun, M., Wang, H., Liu, P., Huang, S., & Fan, P. (2019). A sparse stacked denoising autoencoder with optimized transfer learning applied to the fault diagnosis of rolling bearings. *Measurement*, 146, 305-314.
- VibrAlign, (2019). 3 Types of Mechanical Looseness: What You Need to Know. Retrieved from <https://acoem.us/blog/condition-monitoring/3-types-of-mechanical-looseness-what-you-need-to-know/>
- Xu, H., Song, P., & Liu, B. (2019). A Vibration Signal Anomaly Detection Method Based on Frequency Component Clustering and Isolated Forest Algorithm. *In 2019 IEEE 2nd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, pp. 49-53.

BIOGRAPHIES

Eli Gildish is a data scientist with more than 15 years of experience in applied research, software and algorithm development in the fields of Machine Learning, Artificial Intelligence, Signal Processing and Computer Vision. He is a Senior Algorithm Developer at RSL Electronics and his

current research focuses on application of deep learning to anomaly detection, condition monitoring, diagnostics, and prognostics. Eli studied M. Sc. in Applied Mathematics and B. Sc. in Aerospace Engineering at the Technion – Israel Institute of Technology.

Michael Grebshtein received his Ph.D. degree in Aerospace Engineering from the Technion – Israel Institute of Technology. He is currently Lead diagnostic expert in RSL Electronics Ltd working on innovative PHM solutions including AI-based data analysis and physical models development.

Yehudit Aperia received her Ph.D. degree in mathematics from the Faculty of Mathematics and Computer Science of the Weizmann Institute of Science. She is currently the Head of Intelligent Systems Graduate Program at Afeka Academic College of Engineering in Tel Aviv. Her current research interests include artificial intelligence, applications of deep learning in intelligent systems and reinforcement learning.

Alex Kushnirsky is responsible of Prognostics and Health Management (PHM) and Structural Health Management (SHM) R&D programs at the national and international level. He is a Senior Subject-Matter Expert (SME) Leader of all vibration monitoring programs and HUMS for all rotorcraft platforms in IAF. Alex is responsible for collaboration programs between academy and industry in fields of PHM and SHM in Israel and abroad. He received his B-Tech of mechanical engineering from Ort-Braude Karmiel and MBA degree from College of Management Academic Studies at Rishon Lezion.

Igor Makienko received his M.Sc. degree in Machine Learning and Signal Processing from the Faculty of Electrical Engineering of Technion – Israel Institute of Technology. He is currently Head of PHM group in RSL Electronics Ltd with more than 20 years of experience developing PHM applications where AI-powered Predictive Maintenance Platform for Israeli Air Force being one of them. His current research interests include artificial intelligence and applications of deep learning in PHM.

On the Integration of Fundamental Knowledge about Degradation Processes into Data-Driven Diagnostics and Prognostics Using Theory-Guided Data Science

Simon Hagemeyer¹, Peter Zeiler¹, and Marco F. Huber^{2,3}

¹ *Esslingen University of Applied Sciences, Goepfingen, 73037, Germany*
simon.hagemeyer@hs-esslingen.de
peter.zeiler@hs-esslingen.de

² *Institute of Industrial Manufacturing and Management IFF, University of Stuttgart, 70569 Stuttgart, Germany*

³ *Fraunhofer Institute of Manufacturing Engineering and Automation IPA, 70569 Stuttgart, Germany*
marco.huber@ieee.org

ABSTRACT

In Prognostics and Health Management, there are three main approaches for implementing diagnostic and prognostic applications. These approaches are data-driven methods, physical model-based methods, and combinations of them, in the form of hybrid methods. Each of them has specific advantages but also limitations for their purposeful implementation. In the case of data-driven methods, one of the main limitations is the availability of sufficient training data that adequately cover the relevant state space. For model-based methods, on the other hand, it is often the case that the degradation process of the considered technical system is of significant complexity. In such a scenario physics-based modeling requires great effort or is not possible at all. Combinations of data-driven and model-based approaches in form of hybrid approaches offer the possibility to partially mitigate the shortcomings of the other two approaches, however, require a sufficiently detailed data-driven and physics-based model.

This paper addresses the transitional field between data-driven and hybrid approaches. Despite the issues of formulating a physics-based model that provides a representation of the degradation process, basic knowledge of the considered system and of the laws governing its degradation process is usually available. Integration of such knowledge into a machine learning process is part of a research field that is either called theory-guided data science, (physics) informed machine learning, physics-based learning or physics guided machine learning. First, the state of research in Prognostics and Health

Management on methods of this field is presented and existing research gaps are outlined. Then, a concept is introduced for incorporating fundamental knowledge, such as monotonicity constraints, into data-driven diagnostic and prognostic applications using approaches from theory-guided data science. A special aspect of this concept is its cross-application usability through the consideration of knowledge that repeatedly occurs in diagnostics and prognostics. This is, for example, knowledge about physically justified boundaries whose compliance makes a prediction of the data-driven model plausible in the first place.

1. INTRODUCTION

The choice between a model-based or a data-driven approach is a crucial element of any Prognostics and Health Management (PHM) application. Whether, for example, in the case of condition diagnosis or subsequent prediction of remaining useful life (RUL), the suitability of the respective approach depends on the properties of the particular application. The central prerequisite for a model-based approach is that knowledge on causal relationships of the technical system and its degradation process is available for the formation of a physics-based model. The model-based approach is often characterized by a rather high predictive accuracy and a comparably small amount of required data. However, the utilization of this approach is severely limited by the fact that the degradation processes of many technical systems are of such high complexity that a detailed, purely physics-based modeling is hardly possible (Eker et al., 2016). In addition, such physics-based models are also highly application-specific and therefore have restricted transferability (Byington et al., 2002).

Simon Hagemeyer et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The counterpart to the model-based approaches are the data-driven ones. These originate primarily from the domains of statistics and machine learning. Their implementation requires comparatively small effort; and at least the fundamental learning algorithm has a wide range of applicability (Eker et al., 2016). The methods are based on inductive inference, which underlies the statistical modeling of the training data provided (Huellermeier & Waegeman, 2021). The causal relationships, however, that yield the values of the training data are not learned. Since the data are the only source of information for these methods, they are not suitable for extrapolation into areas with sparse and, in particular, no training data. Accordingly, their purposeful use requires sufficient coverage of the state space by data (Coveney et al., 2016). Furthermore, the lack of comprehension of the causal relationships means that the predictions can take on implausible values that violate fundamental constraints (von Rueden, Mayer, et al., 2021). A behavior that intensifies in areas without training data.

Even though data-driven methods are currently predominant in research on PHM, the lack of data is a major limitation to their widespread industrial application. This affects diagnosis as well as prognosis, which can be subdivided in accordance with Jia et al. (2018) into four tasks:

- **Fault detection:** Detect a fault state/anomaly of a technical system without knowing the root cause. This results in a binary classification problem with the states fault or no fault.
- **Diagnosis:** Assign one or more causes to a detected fault state.
- **Health assessment:** Assess the state of health or the current risk of failure of a system based on its current condition.
- **Prognosis:** Predict the future state of health or RUL.

Each of these tasks involves its own estimation process and is individually affected by the lack of data.

Having a representative data set containing several run-to-failure data sets for each fault mode in each system configuration typically corresponds to a practically impossible amount of effort due to the typical lifetime and variant diversity of many systems. Even the recording of one run-to-failure cycle can take several months or years (Hagmeyer et al., 2021; Hemmer et al., 2019; Pillai et al., 2016). Therefore, Chao et al. (2022) even state that the two aforementioned problems of incomplete physical models and the lack of representative data sets are among the main problems in RUL prediction.

The combination of data-driven and physics-based models is usually referred to as hybrid in PHM. In this context, the term hybrid has a wide range of definitions depending on the literature, as among others Javed et al. (2017), N.-H. Kim et al. (2016), and Liao & Koettig (2014) demonstrate. In this paper, only the combination of entire data-driven and physics-based

models is referred to as a hybrid approach. These offer the possibility to mitigate the limitations of the two approaches described above, but require sufficiently detailed models of both types. In addition, in order to restrict the scope of the following investigations, only approaches in which the data-driven models and incorporated knowledge or physics-based models relate to the same PHM task will be considered. The wide range of approaches to joining models in which they complement each other, for example, by one model doing fault detection and the other doing cause assignment based on it, or by one model doing health estimation and the other describing the degradation progression, is out of the scope of the paper.

Fundamentally, the integration of knowledge into machine learning is a whole research area that has been experiencing a great growth especially in the last five years. Depending on the literature, this research field is referred to as

- *theory-guided data science* see (Karpatne et al., 2017),
- *(physics) informed machine learning* see (von Rueden, Mayer, et al., 2021), (Yucesan & Viana, 2020b),
- *physics-based learning* see (Liu & Goebel, 2018) or
- *physics guided machine learning* see (Rai & Sahu, 2020).

In the following, the term theory-guided data science (TGDS) is used, as Karpatne et al. (2017) were the first to introduce such a designation of the research field. The research area TGDS does not only address the integration of entire physics-based models to increase predictive accuracy in machine learning, but already starts with the integration of knowledge about single principles of the process to be modeled. Here, the term knowledge is used in accordance with von Rueden, Mayer, et al. (2021), in that knowledge is seen as "validated information about relations between entities in certain contexts" (von Rueden et al., 2021).

The topic of this paper, integrating basic knowledge that is not sufficient for holistic modeling, lies in the transition area between data-driven and hybrid. Such basic knowledge already begins with the fact that most technical systems are not capable of self-healing and consequently a predicted degradation curve has to show a monotone increase. However, integration of such knowledge is an aspect that has received comparatively little attention in PHM so far. Although individual approaches have been used in case studies, any overall consideration of their use in PHM is missing. Studies in general, as well as those related to PHM described in the next section, nevertheless already demonstrate the potential of combining data and knowledge for increasing predictive accuracy. Thus, for example, insufficient amounts of data could be compensated. The term predictive accuracy is dependent on the respective PHM task and is evaluated by different metrics, some of which are subject-specific. Typical examples of these metrics are for fault detection *fault detection rate*, for diagnosis *isolation classification rate*, for health assessment *root mean*

squared error, and for prognosis *prognostic horizon* (Saxena et al., 2010; Feldman et al., 2010; Gao et al., 2019).

The purpose of this paper is to take a first step towards a general examination of the use of TGDS in PHM as well as to initiate new research. Therefore, in Section 2 an overview of relevant TGDS approaches that do not require complete physics-based modeling and their employment in PHM is given. Then, in Section 3, concepts of assigning knowledge that occurs across diagnostic and prognostic applications to suitable TGDS methods are introduced. In the last section a conclusion and outlook on future work is given. The overall relevance of the paper's topic for PHM research stems firstly from the fact that the scenario of insufficient training data combined with incomplete physics-based modeling is common for industrial applications of PHM and secondly, that studies already show the potential of TGDS in such scenarios.

2. OVERVIEW OF APPROACHES FOR INTEGRATING KNOWLEDGE INTO MACHINE LEARNING

Already in regular machine learning, knowledge is partially integrated at several places of the learning pipeline. This includes, for example, feature engineering or the selection of the hypothesis set by defining hyperparameter values. TGDS extends the usual building of data-driven models by considering knowledge as a second source of information besides the data (von Rueden, Mayer, et al., 2021). In this paper, methods that do not require entire physics-based models for fusing knowledge and machine learning are considered. This involves knowledge about partial facets of the learning task, such as a subdivision of a problem into subproblems based on physics or knowledge about regularities such as valid bounds of variables, monotonicity conditions, correlations or curve shapes of intermediate and target variables. This form of knowledge integration is characterized by utilizing knowledge about intermediate variables or about valid properties of the target variables. The feature that distinguishes an incorporation of physics-based models from this is that a complete model provides a sufficiently precise estimate of the concrete value of the target variable(s). Thus, the data-driven and the physics-based models provide basically the same kind of information about the target variable, such as the health index (HI) or the RUL information. It is only through this uniformity that the method spectrum of hybrid model ensembles becomes possible.

Literature reviews of TGDS methods already exist, but these are independent of PHM and do not distinguish whether the formation of an entire physics-based model is required, which is highly relevant for PHM due to the complexity of many degradation processes. These PHM-independent works perform a mutually differing distinction of TGDS methods, as shown for example by von Rueden, Mayer, et al. (2021), Aykol et al. (2021), Willard et al. (2020), Karpatne et al. (2017), and

Rai & Sahu (2020). In the following, six approaches are presented that allow the integration of knowledge that does not allow complete modeling. Furthermore, references to the already existing implementations of these approaches in PHM are given.

2.1. Physics-Based Generation of Synthetic Training Data

This method is the most intuitive form of knowledge integration. Here, the available amount of training data is extended by synthetic data points generated on the basis of knowledge. However, the labeling of such data points requires concrete values of the target variable and thus actually a process model. This issue is solved by drawing random samples from the entire range of values of the target variable that are considered valid based on knowledge. This could be data in which the values of the target variable comply with a given set of curves. Even more than when using a physics-based model for labeling, the deviation of the synthetic training data from the correct value is expected to have not only a high variance but also a high bias. Therefore, to improve the accuracy with the data, it is used in a pretraining for a physics-guided initialization instead of being mixed with the regular training data. In the pretraining, the model is trained on a rather simple problem. The actual training based on this, especially with small data sets, serves the subsequent fine-tuning of the machine learning model (Jia et al., 2019).

Several examples for the use of physics-based models to generate synthetic training data exist in PHM, such as Yu et al. (2018) and Sankararaman et al. (2011). However, most of these aim not to improve accuracy but to save computation time in the application phase by replacing the physics-based model with the data-driven one. The enrichment of the training data by knowledge that does not provide a complete modeling has hardly been investigated so far. The authors are so far only aware of Yucesan & Viana (2020a), Yucesan & Viana (2020b), and Dourado & Viana (2019) which apply such pretraining. Based on known correlations of input and target variables, these variables are brought in connection by a hyperplane. Since such linear equations do not correspond to the true hypersurface and in particular since weights are unknown in the equation, random initializations of the weights and thus of the plane are used for the generation of synthetic training data. These paper include just the application of physics-based generation of synthetic training data, but without any investigation on the effect of the pretraining.

2.2. Physics-Based Regularization

The training of a machine learning model is basically an optimization problem. The so-called loss function forms the objective function of the optimization, which evaluates the quality of a hypothesis. The goal of the training is to find a hypothesis that minimizes the loss function. This optimiza-

tion problem can be supplemented by physically based constraints in order to obtain a physically consistent hypothesis as training outcome. The main approach to this is the addition of a special regularization term to the loss function. In regular machine learning, the loss function $L(f)$ mostly consists of a component $\text{loss}(\hat{Y}, Y)$ that captures the agreement of the model output \hat{Y} and the real measured values Y , and a regularization component for constraining the model complexity $R(f)$

$$L(f) = \text{loss}(\hat{Y}, Y) + \lambda \cdot R(f). \quad (1)$$

In physics-based regularization, the loss function is extended by the term $\text{loss}_{phys}(\hat{Y})$, which evaluates whether it satisfies governing physics laws

$$L(f) = \text{loss}(\hat{Y}, Y) + \lambda \cdot R(f) + \gamma \cdot \text{loss}_{phys}(\hat{Y}). \quad (2)$$

Noncompliance with laws is penalized by an increased loss value, which is why, depending on the weighting γ , physically consistent solutions are favored by the training. Since the $\text{loss}_{phys}(\hat{Y})$ is independent of actual measured values, the evaluation of physical conformance is not bound to areas present in the collected data (Muralidhar et al., 2018), (Y. Zhu et al., 2019), (von Rueden, Mayer, et al., 2021). For instance, Muralidhar et al. (2018) introduce equations to embed valid ranges of values by means of rectified linear functions and monotonicity constraints by means of logic operations into the loss function as regularization.

Although this is a relevant approach, a work on the implementation of physics-based regularization in PHM is not known to the authors. However, there is an approach in diagnostic and prognostic applications that can be argued in a wider perspective also as an integration of knowledge and fundamentally shares the same concept. Instead of physics-based knowledge about the degradation process, operational knowledge is incorporated into the loss function. For this purpose, in the case of a regression task instead of a symmetric function such as the squared error an asymmetric function is used for $\text{loss}(\hat{Y}, Y)$. Applied to a RUL prediction, the asymmetric function represents the different costs that arise due to excessive maintenance in the case of RUL underestimation and due to unplanned outages in the case of RUL overestimation. Depending on the application, such a model can be trimmed more towards RUL underestimation or overestimation. The use of such asymmetric loss functions is discussed in the evaluation of several data challenges of the PHM Society as well as by Hoenig et al. (2019), Li et al. (2018), and Saxena et al. (2008).

2.3. Final Hypothesis Set Evaluation

A sufficient generalization of a machine learning model cannot be automatically guaranteed after training. In order to validate training results, extensive test data is usually required,

which is specifically retained from the training. When generating several different models through training, this set of final solution hypotheses cannot only be evaluated using test data, but can also be compared to existing knowledge (von Rueden, Wirtz, et al., 2021). For the evaluation and selection of trained models, both the compliance with individual physics laws and the compliance with physical models can be considered. Even though there is no direct integration of knowledge into the learning process in the final hypothesis set evaluation, the method is still included in the list here because the selection process can lead to better model accuracy in the application phase.

The final hypothesis set evaluation is an intuitive approach that is certainly used regularly in PHM in a basic form. In addition, there are comparable approaches and objectives in explainable machine learning. Based on knowledge, the trained models are analyzed in the so-called post-hoc explanation and assessed with respect to their validity (Burkart & Huber, 2021). One application of the approach is presented by Grezmaek et al. (2019). They show that in a learned model for gearbox diagnosis, the damage frequencies which are most relevant for classification are consistent with knowledge of sideband frequencies.

2.4. Intermediate Physical Variables

The basic idea of this approach is to adapt the hypothesis space by dividing the problem of modeling the relationship between input and target variables into modules based on process knowledge. The inputs and outputs of the modules are thus assigned a physical meaning and, as far as possible, they are related to each other on the basis of knowledge. Thereby on the one hand the problem structure can be considered within the architecture of a single data-driven model, e.g. by adapting the architecture of a neural network and assigning meanings to neurons. On the other hand, for each defined modul an individual data-driven model can also be used (Karpatne et al., 2017), (Willard et al., 2020). If at least a modul can be modeled physics-based in sufficient detail, it is also possible to substitute the respective data-driven model by it. Besides intermediate physical variables, this approach can also be designated for instance as physics-guided architecture or as theory-guided design of model architecture.

This physics-based problem subdivision thus also bridges the gap to knowledge-based feature engineering by in both cases providing information on individual intermediate variables related to the target variable. The goal of this approach is the physics-based subdivision of a problem. However, the ability to incorporate a physics-based model of a subproblem also bridges another gap. This is to the, in the first section excluded hybrid approaches where data-driven and physics-based models are used for different PHM tasks. One such example is the state estimation and the prediction of further

degradation using respectively one of the model types.

In PHM, a physics-based problem subdivision is applied several times, particularly noteworthy here are the same papers as mentioned in physics-based generation of synthetic training data. The idea of incorporating knowledge about the structure of a problem, which is not sufficient for complete modeling, into a data-driven model is applied by Yucesan & Viana (2020a), Yucesan & Viana (2020b), and Dourado & Viana (2019) to the examples of bearing damage in wind turbines and corrosion-influenced material fatigue of aircraft components using recurrent neural networks.

2.5. Auxiliary Task in Multi-Task Learning

Another possibility for the integration of knowledge mentioned by Willard et al. (2020) is the use of multi-task learning. In addition to the actual prediction task, auxiliary tasks are used to estimate related physical variables. These auxiliary tasks are defined based on knowledge of the process and admissible properties of these variables. The unification of both tasks by multi-task learning is intended to leverage their synergy for a more precise as well as physically consistent prediction. It should also be emphasized that the physics-based regularization and auxiliary task in multi-task learning approaches have considerable commonalities. Both shift the position of the optimum, which is searched for during the training process, towards models, which comply with given knowledge. Nevertheless, there is also an affiliation of this approach to intermediate physical variables. The hypothesis set is adjusted by linking related physical variables to the target variable on the basis of knowledge.

In PHM, especially Ozdagli & Koutsoukos (2021) address the use of knowledge about related variables in the context of multi-task learning. The method of employing knowledge-based auxiliary tasks is applied to damage detection in structural health monitoring using neural networks. The labels for the auxiliary tasks are provided in this case by a physics-based model, which, nevertheless, is not fundamentally required for the approach. Compared to the baseline of a purely data-driven neural network, a significant improvement of the classification accuracy is shown. In addition to incorporating knowledge, another advantage of the multi-task approach is the possibility to use labeled data of the additional target variables for training in order to obtain enhanced learning results also for the actual target variable (Caruana, 1997). Examples of such work in PHM include T. S. Kim & Sohn (2020), Chen et al. (2019), and Hinchí & Tkouat (2018). One aspect that is entirely absent in these studies is having knowledge about admissible properties for the related variables and the incorporation of this knowledge into the learning process.

2.6. Knowledge Integration into Probabilistic Graphical Models

The probabilistic graphical models are particularly suitable for the integration of knowledge due to their inherent interpretability. Based on knowledge, nodes and edges can be parameterized, e.g. by specifying an adjacency matrix. As with the multi-task approach, probabilistic graphical models are considered here as a separate case, wherein the integration of knowledge in probabilistic graphical models has already received extensive consideration both in general and in PHM in particular. Depending on the further learning process, this can be seen as an architectural constraint adjusting the hypothesis space in the sense of intermediate variables. If the parameterization of edges represents a priori information that is adapted during training, the learning process is rather guided in one direction in the sense of a regularization. Such ambiguity is also reflected in the different treatment of this approach in the review papers on TGDS mentioned at the beginning of the second section.

The ability to perform knowledge integration of these models is also reflected in the extensive work being done on this at PHM. Liu & Goebel (2018) present a research and development project of the US federal agency National Aeronautics and Space Administration. The goal here is to develop a predictive system that not only assesses the safety status of aircrafts, but of the entire airspace. As a central element of the information fusion, a Bayesian network is used. Juesas et al. (2016) in turn present the integration of imprecise state knowledge into an autoregressive hidden Markov model (ARHMM) using the CMAPSS dataset as a benchmark. The possibility to represent imprecise knowledge allows choosing a compromise between belief and evidence in model generation. Palazuelos et al. (2020) and González et al. (2019) present a graph network where nodes represent the state of system components. An adjacency matrix is used to define connections between nodes of physically related components. The matrix can be learned from data but also created or adapted based on knowledge.

3. CONCEPTION OF A PHM RELATED USE OF TGDS METHODS

Despite the outlined potential of TGDS to improve data-driven diagnostic and prognostic applications, it is also apparent that there are still significant research gaps in this regard. As a first step towards a holistic treatment of the topic, the following sub-sections introduce concepts of assigning knowledge that occurs across diagnostic and prognostic applications to suitable TGDS methods presented in the previous section. The selection of cross-application knowledge is based on the authors' assessment and focuses on knowledge of the degradation process. In PHM, there are also other sources of recurring knowledge related to the degradation process, which are

not considered here. Examples of this include knowledge due to a previous risk assessment such as an FMEA or knowledge about operating conditions.

The basic assumption is that a larger amount of integrated information, whether in the form of knowledge or data, is generally associated with an improvement in the predictive accuracy of a diagnostic or prognostic application. Another assumption is that, although limited in volume, labeled data of the examined degradation process for supervised learning are available in the first place.

In supervised learning, models are trained to reflect the relationship between input and target variables. The structure of the learned model or its information processing to form the estimate of the target variable is not bound to the cause-effect relationships of the modeled process. Consequently, from the authors' point of view, an essential characteristic of knowledge of the modeled process is whether it relates to the target variable that is always present or only to an intermediate variable associated with the target variable that is not inherently included in the model. Hence the following subdivision is provided:

- Concepts for the integration of cross-application knowledge on target variables
- Concepts for the integration of cross-application knowledge on intermediate variables

3.1. Concepts for the Integration of Cross-Application Knowledge on Target Variables

The three TGDS methods that specifically require and incorporate knowledge of the target variable are physics-based generation of synthetic training data (Section 2.1), physics-based regularization (Section 2.2), and final hypothesis set evaluation (Section 2.3).

Knowledge on the curve shape of the degradation process: If the fault mechanism is the same, the shape of the health progression is often identical across applications and therefore known. For example, the fault mechanism determines whether a system is capable of self-healing and thus whether a positive HI gradient is admissible. If this is not the case, a monotone damage progression must be observed. Further examples are the typical convex curves of crack propagation under cyclic loading (Castillo et al., 2010) and in the case of filter clogging the differential pressure increase (Thomas et al., 2001). The latter additionally becoming a linear increase when depth filtration transitions to cake filtration. Even though the level of degradation over time can only be described very imprecisely, there is nevertheless knowledge of shape constraints that should be fundamentally fulfilled in a prediction. Mathematical shape constraints can be well expressed by formulas, which is why physics-based regularization is particularly suitable. Physics-based regularization provides the ability to guide the training into such a direction

that the constraints are met, also for high-dimensional problems. By relying on formulas to express shape constraints, one can also use them to evaluate the final hypothesis set. Although compliance is not enforced in training, it does have an advantage of general applicability especially when considering different types of machine learning methods that involve different loss and training functions.

Knowledge of correlations: If there is no knowledge on strict shape constraints but only on correlations between input variables and target variables, which are not universally met, the physics-based generation of synthetic training data approach is most suitable. The reason for this is that local areas where the synthetic data show a significant bias compared to the actual data can still be adjusted following the pretraining. Other approaches instead would likely result in rather soft constraints or a flawed model as training result with such inaccurate knowledge. As Yucesan & Viana (2020a), Yucesan & Viana (2020b), and Dourado & Viana (2019) demonstrate, engineering estimates on correlations are sufficient to obtain a reasonable initialization of an iteratively trained model by means of a pretraining. Furthermore, for example by Lauer & Bloch (2008) different approaches are presented for also incorporating synthetic training data with different quality than the actual training data in support vector machines with their convex training tasks.

Non-formalized expert knowledge (tacit knowledge): Often, knowledge about degradation processes is available in the form of expert knowledge that is difficult to express in mathematically precise terms. Especially in such cases final hypothesis set evaluation is well suited, since no formulation is required and basic physical correctness of the learned model can be ensured. However, the knowledge-based analysis of trained models is closely related to the topic and problems of interpretable machine learning. The main issue here is the mostly abstract, high-dimensional representation of learned results that are beyond human cognitive comprehension. So, the application of the final hypothesis set evaluation approach requires that the models to be evaluated are intrinsically interpretable or that post-hoc explanations can be applied. Post-hoc explanations require that a low-dimensional and to some extent local representation of the learned behavior can be created without too much loss of information, which can for example be visually perceived. Thereby, significant research gaps in PHM on interpretable machine learning especially on models of time series analysis and prediction applications in general exist (Vollert et al., 2021).

Boundaries of target variables: If, the boundaries of the target variable's permissible range are known, several methods are suitable for ensuring compliance with these boundaries and thus, the basic validity of an estimate. In addition to physically induced boundaries, external requirements, such as a maximum service life of a component, can also yield

Table 1. Summary of recommended methods for integrating knowledge that occurs across applications.

Type of knowledge	Proposed approach for knowledge integration
Knowledge on the curve shape of the degradation process	Physics-Based Regularization (2.2) Final Hypothesis Set Evaluation (2.3)
Knowledge of correlations	Physics-Based Generation of Synthetic Training Data (2.1)
Non-formalized expert knowledge	Final Hypothesis Set Evaluation (2.3)
Boundaries of target variables	Physics-Based Regularization (2.2) Final Hypothesis Set Evaluation (2.3) Probabilistic Graphical Model (2.6) Intermediate Physical Variables (2.4)
Knowledge of the problem structure	Intermediate Physical Variables (2.4) Probabilistic Graphical Model (2.6)
Knowledge and extensive data of intermediate variables	Auxiliary Task in Multi-Task Learning (2.5) Intermediate Physical Variables (2.4) Probabilistic Graphical Model (2.6)

such boundaries. Besides physics-based regularization and final hypothesis set evaluation, the approaches of using graphical models and intermediate physical variables can also be utilized to enforce compliance with such boundaries. With graphical models, edges can be parameterized accordingly. In the case of the intermediate physical variables approach, a model structure that fundamentally ensures the compliance can be specified, in simple cases of constant boundaries already by the choice of an output function.

3.2. Concepts for the Integration of Cross-Application Knowledge on Intermediate Variables

The three methods that can be used also in case of knowledge of the problem structure and intermediate variables are intermediate physical variables (Section 2.4), auxiliary task in multi-task learning (Section 2.5), and probabilistic graphical models (Section 2.6). In the following, a distinction is made between only two cases.

Knowledge of the problem structure: If knowledge about the structure of a problem and about relevant intermediate variables is available, a physically based subdivision according to the approach intermediate physical variables can usually be applied. The same holds for graphical models whose nodes can be assigned a meaning and also edges can be specified accordingly between nodes. Especially if the data-driven estimation of intermediate variables is considered as a learning task on its own, the concepts described above for the integration of knowledge about target variables can be applied to this subproblem. That knowledge about a problem’s structure and intermediate variables is often available is shown by the extensive work on hybrid methods where different model types take over individual subtasks (Eker et al., 2019). A further evidence is the physics-based feature engineering already mentioned in Section 2, where also extensive work is done, especially on rotating systems like rolling bearings or gears (J. Zhu et al., 2014).

Knowledge and extensive data of intermediate variables:

Multi-task learning addresses among others the case when, in addition to knowledge of intermediate variables, extensive labeled data on these variables is also available. Instead of learning to assign known intermediate variables as a submodule or using probabilistic graphical models, multi-task learning can alternatively use them as additional target variables. Although there is still a considerable need for research on the integration of knowledge in multi-task learning, the approach of using additional labeled data that do not include the actual target variable already offers great potential. In accordance with T. S. Kim & Sohn (2020), the estimation of the current HI can form an auxiliary task in a prediction application. From the authors’ point of view, this approach is of high relevance, since it allows data to be used for learning a prognosis model, which do not contain any health change and thus are of minor use in a regular prognosis application. In many applications with long test durations, such as ball bearings, tests with predamaged components on fixed fault conditions and therefore health are common practice (Chen et al., 2018). With multi-task learning and knowledge of such related variables, this kind of test data can also be used for prognosis development.

A summary of the concepts for assigning knowledge types and TGDS methods is given in Table 1.

4. CONCLUSIONS AND OUTLOOK

There are three approaches to realizing diagnostic and prognostic tasks. In this paper, at first, these approaches are characterized. Thereby, connections between hybrid methods and the research field of TGDS can be identified. Subsequently, main aspects of TGDS are introduced and the potential of TGDS in PHM is outlined. The focus here is on methods for the integration of knowledge in machine learning, which do not require complete physics-based models, but rather knowledge of individual properties of the degradation process. For

this purpose, a definition for the designation model is given. The presented overview of the relevant TGDS approaches illustrates them in detail and also points out studies on PHM that already employ them. In doing so, several research gaps can be identified. Based on the overview, cross-application knowledge occurring in diagnostics and prognostics is stated and concepts for integrating it into the learning process are proposed. The description of suitable methods contained therein is based primarily on theoretical considerations and, where available, on transferable findings from other work.

Overall, the paper makes an initial contribution to a holistic investigation of the incorporation of knowledge into machine learning in diagnostics and prognostics. There is significant potential for the use of TGDS in PHM, but also a great need for further research. Concerning the latter, on the one hand, there is much more knowledge for which a procedure for the integration is of cross-application benefit. On the other hand, the given theoretical concepts have to be investigated more thoroughly, supplemented by PHM-specific aspects such as uncertainty considerations, and verified by empirical studies. In addition, overlapping research fields such as transfer learning and fuzzy machine learning have to be considered, where the integration of knowledge is also a partial aspect.

REFERENCES

- Aykol, M., Gopal, C. B., Anapolsky, A., Herring, P. K., van Vlijmen, B., Berliner, M. D., ... Storey, B. D. (2021). Perspective—combining physics and machine learning to predict battery lifetime. *Journal of The Electrochemical Society*, 168(3). doi: 10.1149/1945-7111/abec55
- Burkart, N., & Huber, M. F. (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70, 245–317. doi: 10.1613/jair.1.12228
- Byington, C. S., Roemer, M. J., & Galie, T. (2002). Prognostic enhancements to diagnostic systems for improved condition - based maintenance. In *Proceedings of the IEEE aerospace conference*. Big Sky, MT, USA. doi: 10.1109/AERO.2002.1036120
- Caruana, R. (1997). Multitask learning. *Machine Learning*, 28(1), 41–75. doi: 10.1023/a:1007379606734
- Castillo, C., Fernández-Canteli, A., Castillo, E., & Pinto, H. (2010). Building models for crack propagation under fatigue loads: application to macrocrack growth. *Fatigue & Fracture of Engineering Materials & Structures*, 33(10), 619–632. doi: 10.1111/j.1460-2695.2010.01475.x
- Chao, M. A., Kulkarni, C., Goebel, K., & Fink, O. (2022). Fusing physics-based and deep learning models for prognostics. *Reliability Engineering & System Safety*, 217. doi: 10.1016/j.ress.2021.107961
- Chen, Y., Peng, G., Xie, C., Zhang, W., Li, C., & Liu, S. (2018). ACDIN: Bridging the gap between artificial and real bearing damages for bearing fault diagnosis. *Neurocomputing*, 294, 61–71. doi: 10.1016/j.neucom.2018.03.014
- Chen, Y., Zhang, C., Zhang, N., Chen, Y., & Wang, H. (2019). Multi-task learning and attention mechanism based long short-term memory for temperature prediction of EMU bearing. In *Proceedings of the prognostics and system health management conference (PHM-qingdao)*. Qingdao, China: IEEE. doi: 10.1109/phm-qingdao46334.2019.8942914
- Coveney, P. V., Dougherty, E. R., & Highfield, R. R. (2016). Big data need big theory too. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2080), 20160153. doi: 10.1098/rsta.2016.0153
- Dourado, A., & Viana, F. A. C. (2019). Physics-informed neural networks for corrosion-fatigue prognosis. In *Proceedings of the annual conference of the phm society* (Vol. 11). Scottsdale, Arizona, USA.
- Eker, O. F., Camci, F., & Jennions, I. K. (2016). Physics-based prognostic modelling of filter clogging phenomena. *Mechanical Systems and Signal Processing*, 75, 395–412. doi: 10.1016/j.ymsp.2015.12.011
- Eker, O. F., Camci, F., & Jennions, I. K. (2019). A new hybrid prognostic methodology. *International Journal of Prognostics and Health Management*, 11(2).
- Feldman, A., Kurtoglu, T., Narasimhan, S., Poll, S., Garcia, D., Kleer, J. D., ... Gemund, A. V. (2010). Empirical evaluation of diagnostic algorithm performance using a generic framework. *International Journal of Prognostics and Health Management*, 1. doi: 10.36001/ijphm.2010.v1i1.1344
- Gao, T., Lu, C., & Hao, M. (2019). Design requirements of PHM system fault diagnosis capability. In *Proceedings of the 2019 chinese automation congress (CAC)*. Hangzhou, China: IEEE. doi: 10.1109/cac48633.2019.8996303
- González, I., Cáceres, J., Droguett, E. L., & López-Campos, M. (2019). Graph convolutional networks for health state diagnostics. In *Proceedings of the 29th european safety and reliability conference* (p. 1208-1213). Hannover, Germany.
- Grezmak, J., Wang, P., Sun, C., & Gao, R. X. (2019). Explainable convolutional neural network for gearbox fault diagnosis. *Procedia CIRP*, 80, 476–481. doi: 10.1016/j.procir.2018.12.008
- Hagmeyer, S., Mauthe, F., & Zeiler, P. (2021). Creation of publicly available data sets for prognostics and diagnostics addressing data scenarios relevant to industrial applications. *International Journal*

- of *Prognostics and Health Management*, 12(2). doi: <https://doi.org/10.36001/ijphm.2021.v12i2.3087>
- Hemmer, M., Klausen, A., van Khang, H., Robbersmyr, K. G., & Waag, T. I. (2019). Simulation-driven deep classification of bearing faults from raw vibration data. *International Journal of Prognostics and Health Management*.
- Hinchi, A. Z., & Tkiouat, M. (2018). A multi-task deep learning model for rolling element-bearing diagnostics. In *Proceedings of the european conference of the phm society* (Vol. 4). Philadelphia, Pennsylvania, USA.
- Hoening, M., Hagemeyer, S., & Zeiler, P. (2019). Enhancing remaining useful lifetime prediction by an advanced ensemble method adapted to the specific characteristics of prognostics and health management. In *Proceedings of the 29th european safety and reliability conference* (pp. 1155–1162). Hannover, Germany.
- Huellermeier, E., & Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110(3), 457–506. doi: 10.1007/s10994-021-05946-3
- Javed, K., Gouriveau, R., & Zerhouni, N. (2017). State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels. *Mechanical Systems and Signal Processing*, 94, 214–236. doi: 10.1016/j.ymsp.2017.01.050
- Jia, X., Huang, B., Feng, J., Cai, H., & Lee, J. (2018). Review of phm data competitions from 2008 to 2017: Methodologies and analytics. In *Proceedings of the annual conference of prognostics and health management society* (Vol. 10). Philadelphia, Pennsylvania, USA.
- Jia, X., Willard, J., Karpatne, A., Read, J., Zwart, J., Steinbach, M., & Kumar, V. (2019). Physics guided RNNs for modeling dynamical systems: A case study in simulating lake temperature profiles. In *Proceedings of the 2019 SIAM international conference on data mining* (pp. 558–566). Calgary, Kanada. doi: 10.1137/1.9781611975673.63
- Juesas, P., Ramasso, E., Drujont, S., & Placet, V. (2016). On partially supervised learning and inference in dynamic bayesiannetworks for prognostics with uncertain factual evidence: illustration with markov switching models. In *Third european conference of the phm society*. Bilbao, Spanien.
- Karpatne, A., Atluri, G., Faghmous, J. H., Steinbach, M., Banerjee, A., Ganguly, A., ... Kumar, V. (2017). Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering*, 29(10), 2318–2331. doi: 10.1109/tkde.2017.2720168
- Kim, N.-H., An, D., & Choi, J.-H. (2016). *Prognostics and health management of engineering systems: An introduction*. Springer International Publishing.
- Kim, T. S., & Sohn, S. Y. (2020). Multitask learning for health condition identification and remaining useful life prediction: deep convolutional neural network approach. *Journal of Intelligent Manufacturing*, 32(8), 2169–2179. doi: 10.1007/s10845-020-01630-w
- Lauer, F., & Bloch, G. (2008). Incorporating prior knowledge in support vector machines for classification: A review. *Neurocomputing*, 71(7-9), 1578–1594. doi: 10.1016/j.neucom.2007.04.010
- Li, X., Ding, Q., & Sun, J.-Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172, 1–11. doi: 10.1016/j.res.2017.11.021
- Liao, L., & Koettig, F. (2014). Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. *IEEE Transactions on Reliability*, 63(1), 191–207. doi: 10.1109/TR.2014.2299152
- Liu, Y., & Goebel, K. (2018). Information fusion for national airspace system prognostics: A nasa uli project. In *Proceedings of the annual conference of the phm society* (Vol. 10). Philadelphia, Pennsylvania, USA.
- Muralidhar, N., Islam, M. R., Marwah, M., Karpatne, A., & Ramakrishnan, N. (2018). Incorporating prior domain knowledge into deep neural networks. In *Proceedings of the IEEE international conference on big data (big data)*. Seattle, WA, USA. doi: 10.1109/bigdata.2018.8621955
- Ozdagli, A. I., & Koutsoukos, X. (2021). Model-based damage detection through physics guided learning. In *Proceedings of the proceedings of the annual conference of the PHM society* (Vol. 13). Online: PHM Society. doi: 10.36001/phmconf.2021.v13i1.3012
- Palazuelos, A. R.-T., Droguett, E. L., & Groth, K. M. (2020). A system-level prognostics and health management framework based on graph convolutional neural networks. In *Proceedings of the 30th european safety and reliability conference and 15th probabilistic safety assessment and management conference*. Venice, Italy, (online). doi: 10.3850/978-981-14-8593-0_4165-cd
- Pillai, P., Kaushik, A., Bhavikatti, S., Roy, A., & Kumar, V. (2016). A hybrid approach for fusing physics and data for failure prediction. *International Journal of Prognostics and Health Management*, 7(4).
- Rai, R., & Sahu, C. K. (2020). Driven by data or derived through physics? a review of hybrid physics guided machine learning techniques with cyber-physical system (CPS) focus. *IEEE Access*, 8, 71050–71073. doi: 10.1109/access.2020.2987324
- Sankararaman, S., Ling, Y., Shantz, C., & Mahadevan, S. (2011). Uncertainty quantification in fatigue crack growth prognosis. *International Journal of Prognostics and Health Management*, 2(1).

- Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008). Metrics for evaluating performance of prognostic techniques. In *Proceedings of the international conference on prognostics and health management*. Denver, CO, USA: IEEE. doi: 10.1109/phm.2008.4711436
- Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and Health Management, 1*. doi: 10.36001/ijphm.2010.v1i1.1336
- Thomas, D., Penicot, P., Contal, P., Leclerc, D., & Vendel, J. (2001). Clogging of fibrous filters by solid aerosol particles experimental and modelling study. *Chemical Engineering Science, 56*(11), 3549–3561. doi: 10.1016/s0009-2509(01)00041-0
- Vollert, S., Atzmueller, M., & Theissler, A. (2021). Interpretable machine learning: A brief survey from the predictive maintenance perspective. In *Proceedings of the IEEE international conference on emerging technologies and factory automation (ETFA)*. Vasteras, Sweden: IEEE. doi: 10.1109/etfa45728.2021.9613467
- von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Gieselbach, S., Heese, R., ... Schuecker, J. (2021). Informed machine learning - a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*. doi: 10.1109/tkde.2021.3079836
- von Rueden, L., Wirtz, T., Hueger, F., Schneider, J. D., Pitakowski, N., & Bauckhage, C. (2021). Street-map based validation of semantic segmentation in autonomous driving. In *Proceedings of the international conference on pattern recognition (ICPR)*. Milan, Italy: IEEE. doi: 10.1109/icpr48806.2021.9413292
- Willard, J., Jia, X., Xu, S., Steinbach, M., & Kumar, V. (2020). Integrating scientific knowledge with machine learning for engineering and environmental systems. *arXiv: 2003.04919v6*.
- Yu, Y., Yao, H., & Liu, Y. (2018). Physics-based learning for aircraft dynamics simulation. In *Proceedings of the annual conference of the phm society* (Vol. 10). Philadelphia, Pennsylvania, USA. doi: 10.36001/phm-conf.2018.v10i1.513
- Yucesan, Y. A., & Viana, F. (2020a). Hybrid model for wind turbine main bearing fatigue with uncertainty in grease observations. In *Proceedings of the annual conference of the PHM society* (Vol. 12). online. doi: 10.36001/phm-conf.2020.v12i1.1139
- Yucesan, Y. A., & Viana, F. A. C. (2020b). A physics-informed neural network for wind turbine mainbearing fatigue. *International Journal of Prognostics and Health Management, 11*. doi: 10.36001/ijphm.2020.v11i1.2594
- Zhu, J., Nostrand, T., Spiegel, C., & Morton, B. (2014). Survey of condition indicators for condition monitoring systems. In *Proceedings of the annual conference of prognostics and health management society* (Vol. 6). Fort Worth, Texas, USA.
- Zhu, Y., Zabarar, N., Koutsourelakis, P.-S., & Perdikaris, P. (2019). Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics, 394*, 56–81. doi: 10.1016/j.jcp.2019.05.024

Toward Runtime Assurance of Complex Systems with AI Components

Yuning He¹, Johann Schumann², Huafeng Yu³

¹ NASA, NASA Ames Research Center, Moffett Field, CA, 94035, USA
Yuning.He@nasa.gov

² KBR/Wyle, NASA Ames Research Center, Moffett Field, CA, 94035, USA
Johann.M.Schumann@nasa.gov

³ Boeing Research & Technology, Huntsville, AL 35808, USA
huafeng.yu@boeing.com

ABSTRACT

AI components (e.g., Deep Neural Networks) are increasingly used in safety-relevant aerospace applications. Rigorous Verification and Validation (V&V) is mandatory for such components, yet V&V techniques for DNNs are still in their infancy and can often only provide relatively weak guarantees. In this paper, we will present a runtime-monitoring architecture, which combines the advanced statistical analysis framework SYS AI (System Analysis using Statistical AI) with temporal and probabilistic runtime monitoring carried out by R2U2 (Realizable, Responsive, and Unobtrusive Unit). We will present initial results of our tool set and architecture on a case study, a DNN-based autonomous centerline tracking system (ACT).

1. INTRODUCTION

Artificial Intelligence (AI) components such as Deep Neural Networks (DNNs) have found their way into many complex systems in the aerospace and automotive domain. Such use of AI exhibits tremendous benefits, but most of the applications are safety-critical, and failures might lead to loss of vehicle and mission or even to loss of life. Certification standards for safety-critical systems (e.g., DO-178C or ISO 26262) require processes with rigorous Verification and Validation (V&V) goals. However, techniques for V&V for AI components are still in their infancy. Certification standards for safety-critical components, which are based upon AI and machine learning are still under development (e.g., (EASA, 2021; He, Yu, Brat, & Davies, 2022)).

Yuning He et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The intended target applications for AI and machine learning systems also require that such systems need to operate properly under a wide variety of different operational and environmental conditions, as well as under failures. This is, in particular, true for the area of autonomous vehicles, where AI components are taking over tasks of perception and decision making.

These tasks also require that failures, abnormal environmental conditions, and other hazards can be detected in real time, are properly diagnosed, and potential mitigation actions are proposed to the decision-making layers.

In order to facilitate a safe operation of the complex system and potentially support certification, advanced runtime monitoring is essential. Inspired by the ASTM-F3269 (ASTM, Nov 2021) standard, which defines a runtime assurance architecture, we propose a runtime architecture, which

- uses efficient and advanced runtime monitoring techniques (temporal logic, Bayesian probabilistic reasoning) provided by the R2U2 system, and synergistically combine it with the
- SYS AI analysis framework for complex systems with AI components.

In the architecture, which will be presented in this paper, the R2U2 system dynamically monitors numerous system signals and information originating from the AI component. Temporal logic observers for past and mission time temporal logic make it possible to check a multitude of complex properties, which need to be fulfilled when the complex system is working properly. If the AI system is not working as expected or failures occur, the R2U2 monitors and reasoners provide diagnostic information, which can be used to operate a "runtime assurance switch", which causes to activate safe (and potentially verified) fall-back components.

A complex (AI) system requires complex properties and parameters to be checked; simple thresholding is, in most cases, not sufficient. But how can those complex properties and parameters be obtained?

For this task, we use SYS AI (System Analysis using Statistical AI), our flexible statistical learning framework for V&V and analysis of complex and high-dimensional cyber-physical systems with AI components. SYS AI provides algorithms to efficiently create statistical models, perform safety-envelope analysis, characterize safety boundaries, and carry out time series analysis. SYS AI is used during design and V&V time of the system development process. Learned statistical models of the complex system and its AI components, which are produced by SYS AI during V&V provide the detailed information that is necessary to enable the R2U2 runtime monitor to efficiently perform advanced safety and performance checks for nominal and off-nominal conditions. These checks are expressed as temporal properties and also include Bayesian statistical reasoning.

In this paper, we propose a draft of a process that uses SYS AI for system analysis and feedback to the designer during development time and that transfers essential information to be used by the R2U2 observers in our runtime monitoring architecture.

We will demonstrate our approach with a case study on a DNN-based autonomous centerline tracking system (ACT). This ACT system uses a vision-based deep neural network to guide an aircraft down the runway during taxi. We will illustrate the capabilities of this architecture using safety-boundary monitoring and handling of a class of camera-related failures.

The rest of the paper is structured as follows: Sections 2 and 3 present background about our SYS AI statistical framework and the R2U2 tool, respectively. In Section 4, we will in detail describe our monitoring architecture, which is based upon R2U2 and define a process, on how SYS AI can provide system model data and parameters, which are needed for the monitor. Section 5 focuses on our case study on autonomous centerline tracking (ACT). We first describe experiments on monitoring the system under nominal operating conditions and then illustrate the capabilities of our architecture on a selected failure case: partial obstruction of the camera by dirt or an insect on the camera lens. Section 6 presents related work and Section 7 concludes and discusses future work.

2. BACKGROUND: THE STATISTICAL ANALYSIS FRAMEWORK SYS AI

SYS AI (System Analysis using Statistical AI) is a flexible statistical learning framework for V&V and analysis of complex and high-dimensional cyber-physical systems with AI components. Figure 1 shows the high-level architecture of SYS AI analysis framework. On the left-hand side, we have

the “system under test” (SuT), which in our case is the ATC system and the XPlane simulator, as described in the previous section. The SuT is executed given a set of parameters and initial conditions provided by the statistical learning model of SYS AI. The result of the test run, which could be a binary safe/not-safe information, a single value (e.g., $ct_{e_{max}}$), or an entire time series is provided back to SYS AI. These data are then used to incrementally construct our statistical model.

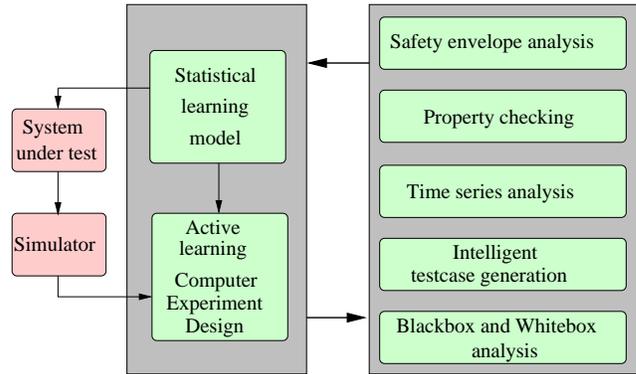


Figure 1. SYS AI architecture

The interface between SYS AI and the SuT is designed to be very small and generic, so that systems implemented in R, Matlab, Java, or Python can be connected easily. For the representation and construction of the statistical SYS AI model, we are using Dynamic Regression Trees (DynaTrees (Taddy, Gramacy, & Polson, 2011; Gramacy & Polson, 2011)), a dynamic Gaussian process model based upon Particle Filters. DynaTrees are regression and classification learning models with complicated response surfaces in on-line application settings. DynaTrees create a sequential tree model whose state changes over time with the accumulation of new data, and provide particle learning algorithms that allow for the efficient on-line posterior filtering of tree-states. A major advantage of DynaTrees is that they allow for the use of very simple models within each partition. The models also facilitate a natural division in sequential particle-based inference: tree dynamics are defined through a few potential changes that are local to each newly arrived observation, while global uncertainty is captured by the ensemble of particles.

This surrogate model is initialized with available training data and incrementally refined using candidate data points that are produced by our active learning module. It evaluates the current surrogate model using a customized active-learning heuristics and suggests candidate data points that provide most information for model refinement. For these candidate points, the ground truth is obtained by executing the SuT.

SYS AI features customizable heuristics that allow the active learning to focus on particular characteristics of the model. Classical algorithms like ALM (MacKay, 1992) or ALC (Cohn,

1996) focus on under-explored regions in general of the domain space. Inspired by (Jones, Schonlau, & Welch, 1998) and work on contour finding algorithms, we loosely follow (Ranjan, Bingham, & Michailidis, 2008) and define our boundary-aware metric boundary-EI (He, 2015, 2012) that puts the focus of the search into “interesting” and potentially “troublesome” areas near safety boundaries. Here, our surrogate model therefore exhibits substantially more details than in other areas that are not of interest. This exploration is guided by the selected active learning heuristics and is able to cover the entire input space with a low number of data points.

The SYS-IAI framework and the underlying models and algorithms are described in detail in (He & Schumann, 2020). SYS-IAI has been used for the analysis of several complex and safety-critical aerospace systems (He, 2015; He et al., 2022; He, Yu, Brat, & Davies, 2021).

3. BACKGROUND: R2U2

The R2U2 (Realizable, Responsive, and Unobtrusive Unit) (Roziar & Schumann, 2017; Reinbacher, Roziar, & Schumann, 2014; Geist, Roziar, & Schumann, 2014) is an on-board monitoring system to continuously monitor system and safety properties of a cyber-physical system or its components. Health models within this framework (Schumann, Roziar, et al., 2015) are defined using Metric Temporal Logic (MTL) and Mission-time Linear Temporal Logic (LTL) (Reinbacher et al., 2014) for expressing temporal properties as well as Bayesian Networks (BN) for probabilistic and diagnostic reasoning. A signal processing unit reads in continuous sensor signals or information from the prognostics unit and performs filtering and discretization operations. Figure 2 shows the high-level architecture of R2U2.

A large number of safety and performance properties for ACT can be formulated using temporal logic. Some properties directly monitor the DNN component, e.g., a simple range check for the DNN outputs, e.g. $\square(|cte_{NN}| < 30)$. With these instantaneous properties, which have no temporal component, the current behavior of the DNN as well as the aircraft (e.g., the commanded steering angle for the front-wheel shall be limited). Proper temporal formulas are used to suppress short dropouts and deviations of the DNN output, as they will be counter-acted by the ACT controller. Of more interest are temporal properties, which limit the number of bad outputs per minute, or classification of longer-duration problems. For example,

$$\square((|he| > 10^\circ) \mathcal{U}_{[0s,9s]}(|he| \leq 8^\circ)) \quad (1)$$

raises an alarm, if a large heading error persists for more than 10 seconds, indicating a possibly unbounded movement to the edge of the runway. In our application for the dynamic comparison of DNN performance, which will be described below, we use temporal properties to analyze a short temporal

trace of the DNN and to analyze closed-loop behavior of the ATC.

On the system level, R2U2 can be used, for example, to continuously check for oscillations occurring in ATC, as they might cause poor performance or can lead to unsafe situations. For the definition of all temporal operators and more examples see (Roziar & Schumann, 2017; Schumann, Roychoudhury, & Kulkarni, 2015).

R2U2 can also perform efficient Bayesian reasoning and has a built-in model-based prognostics engine, which will be helpful for monitoring an AI-based system, but these capabilities have not yet been used for this paper.

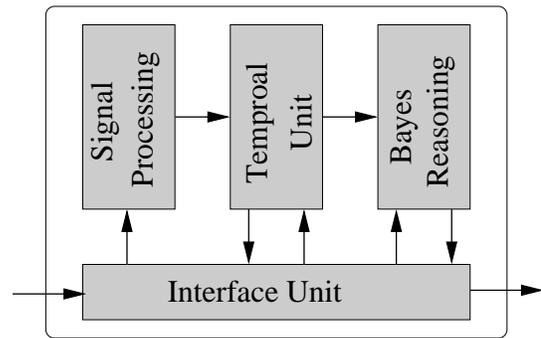


Figure 2. R2U2 architecture with major components for signal processing including prognostics, and temporal and Bayesian reasoning

4. MONITORING AND ASSURANCE ARCHITECTURE

In this section, we present our architecture for monitoring of the complex AI component, using the R2U2 runtime monitor. Important information for the R2U2 properties are produced by SYS-IAI during statistical analysis of the system at design and V&V time. For a synergistic combination of both tools, we propose a draft of a process.

The main goal of this architecture is to provide a framework for the monitoring of a complex AI system, e.g., a Deep Neural Network, during runtime. Future work (see Section 7) will refine that architecture into a runtime assertion framework suitable for certification purposes.

4.1. Monitoring Architecture

The R2U2 system dynamically monitors numerous signals and information provided by the system or its components. It can provide Boolean results on any violation, but can also perform Bayesian reasoning, returning probabilities and confidence values.

For the continuous monitoring of an AI component in a potentially safety-critical system, we have designed, inspired by the ASTM F-3269 RTA (ASTM, Nov 2021), an architecture

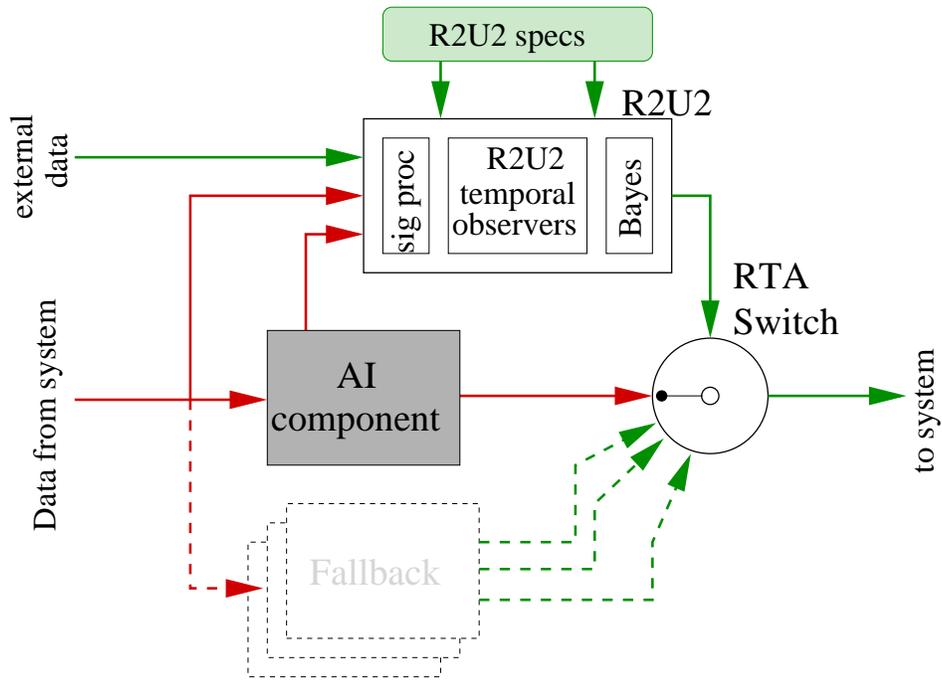


Figure 3. R2U2 runtime monitoring architecture (inspired by (Nagarajan et al., 2021), Fig. 1)

as shown in Figure 3.

The complex AI component, e.g., a Deep Neural Network, is shown as a gray box in the center of the figure. It receives inputs from the system, e.g., camera inputs or sensor signals and processes them. The results (e.g., estimated position of the AC on the runway) are then passed through the RTA switch back to the system, e.g., the aircraft controller. In nominal operations, the RTA switch is set to route the signals from the AI component to the system.

In parallel, R2U2 receives the system signals, as well as signals from the AI component. The latter signals can be, for example, output of the neural network, confidence values for the output, or internal values. The latter, for example, would be important in applications, where the Neural Network is trained or adapted during operation.

In addition to these signals, R2U2 can receive external data of high integrity (e.g., pilot input, redundant sensors, etc). The properties for R2U2 and their parameters have been designed and augmented with results from the SYS AI analysis as discussed below. R2U2 is operating on inputs and specifications and produces an updated result every time step. Typically, R2U2 is operated with a rate of 1Hz or 10Hz. The R2U2 output is used to control the RTA switch: in case, R2U2 detects a violation of important safety/performance properties, the RTA switch can be turned to use a fallback component instead of the AI component to retain system safety and (at least limited) performance. Multiple fallback methods might be provided, ranging from algorithmic components (e.g., sim-

ple dead reckoning) to entering a fail-safe mode, stopping the AC, and contact a remote operator.

4.2. Development Process

Figure 4 illustrates the overall development and monitoring process. Based upon detailed system requirements, the system with AI components is developed and the DNN(s) are trained using training data. At this V&V stage, SYS AI can be used for analysis of training data, characterization of safety regions in a high-dimensional state space, as well as analysis of the system’s behavior under failures (He et al., 2022, 2021). Analysis results also provide feedback to the designer.

After system development and testing, the system is being deployed. At this stage, the R2U2 runtime monitoring is active while the system is in operation. Without affecting the overall system behavior (unobtrusiveness), a multitude of temporal and probabilistic properties can be checked and warning signals or alarms be generated. The statistical models and results, produced by SYS AI, are used to define and customize properties to be checked by R2U2 (vertical red arrow). The information passed can range from simple threshold parameters, whose values have been determined by SYS AI’s safety-boundary characterization. In that case, SYS AI’s advanced capabilities for the geometric characterization of safety boundaries can be used for setting up efficient R2U2 property checking.

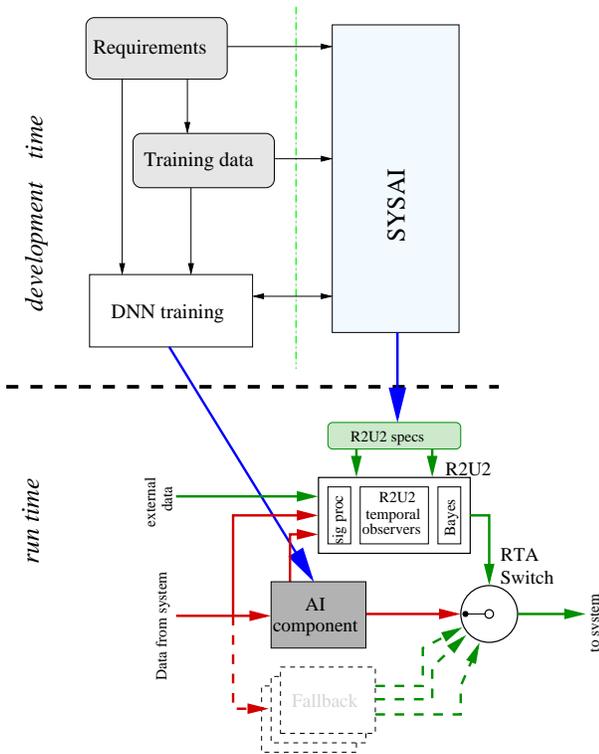


Figure 4. Tool chain and process for the combination of SYSAI and R2U2

5. CASE STUDY

5.1. Autonomous Centerline Tracking

As a case study for our approach, we use the ACT (Autonomous Center Line Tracking) system, which enables autonomous taxiing, one of the most important ground operations for Unmanned Aerial Systems. The core component of ACT is a Deep Neural Network (DNN) that takes images as inputs from cameras mounted on the aircraft’s wings (Figure 5). The DNN component continuously estimates the position and orientation of the aircraft with respect to the runway center line. These values are the cross-track error cte in meters, and the heading error he in degrees, respectively.

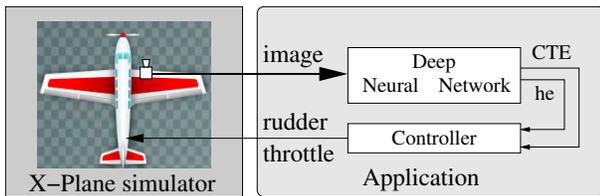


Figure 5. Architectural Overview of ATC

A simple fixed-gain proportional controller uses this information to produce control signals to steer the aircraft left and right. A separate controller keeps the aircraft is rolling with a

constant, low speed.

For our experiments, the X-Plane Flight Simulator¹ is used as simulation environment. A simulated camera takes information from the simulator display; the control signals for throttle and rudder are sent to the X-Plane simulator using the programmatic interface NASA XPlane Connect.²

The DNN is a multi-layer feed-forward network with ReLU nodes. The DNNs are implemented using the TensorFlow framework³ and have been trained on data that have been obtained with the simulated aircraft within the X-Plane simulator. Note that in this application, DNN is not learning any time-series data. Rather the DNN is learning a mapping between the input image (showing a part of the runway) and the corresponding cte and he values.

5.2. Nominal Safety Regions

For the setup of the R2U2 specifications for nominal operation, it is, among others, important to establish reasonable safety thresholds for the neural network outputs using SYSAI. As described above, the ATC DNN produces two outputs, the cross-track error CTE and the heading error he . SYSAI can perform a simultaneous analysis for both parameter, but for this paper we focus on CTE to simplify the presentation of results. During a ATC-guided run, the value of CTE must not surpass the safety threshold θ_{CTE} , i.e., our safety condition is $CTE < \theta_{CTE}$. Obviously, if θ_{CTE} is very small, only few runs will be successful and most runs will violate our threshold safety property.

On the other hand, a large threshold would allow almost all runs to succeed, but the aircraft might veer off the runway proper, which is an unsafe situation. We also have to assume that the AC does not always start exactly at the beginning of the runway precisely on the center line and is perfectly assigned to the center line. Rather, the initial conditions imply non-zero initial cross track error CTE_0 and he_0 . With out SYSAI analysis, we want to find out (a) what are the success rates for a given threshold, and within which geometric boundaries of the AC initial position and heading, a good success rate can be accomplished. In this experiment, we therefore allow SYSAI to vary the initial parameters CTE_0 and he_0 .

Figure 6 shows how, for a given threshold, the starting position and heading of the AC influences the success of a run. Each dot in each panel indicates the starting position of the AC, the protruding line shows the initial AC heading. In each panel, the initial part of the runway is shown, going from the lower left to the upper right. If the threshold is very low, almost no runs are successful (Figure 6A). A somewhat larger

¹www.xplane.com

²<https://github.com/nasa/XPlaneConnect>

³<https://www.tensorflow.org>

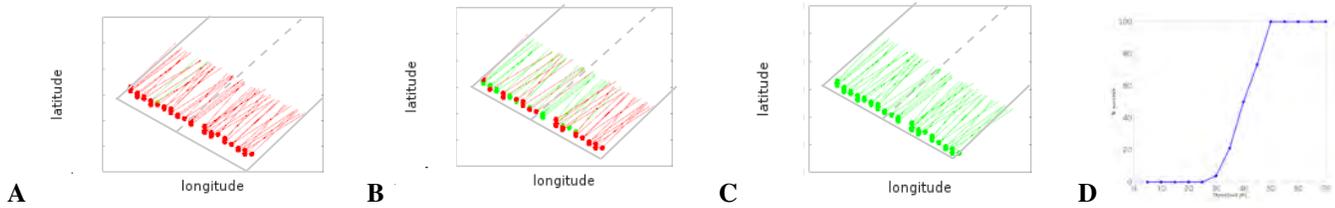


Figure 6. Threshold analysis: success (green) or failure (red) for different initial positions and headings of the AC at the beginning of the runway for different thresholds (A: 30ft, B: 40ft, and C: 50ft). The dots mark the initial position, the lines indicate the heading of the AC. D: success rate (in %) over threshold.

threshold (Figure 6B) shows that around half of the runs are successful. Here it can be seen that the starting position actually makes a difference: starting positions to the left of the center line tend to be much more successful than when starting on the right of the center line. This result can be a basis for further analysis of the coverage of training data, camera placement, or the controller design. Finally, when the threshold is very large, all runs succeed. Figure 6D shows the success rate (in %) for different thresholds.

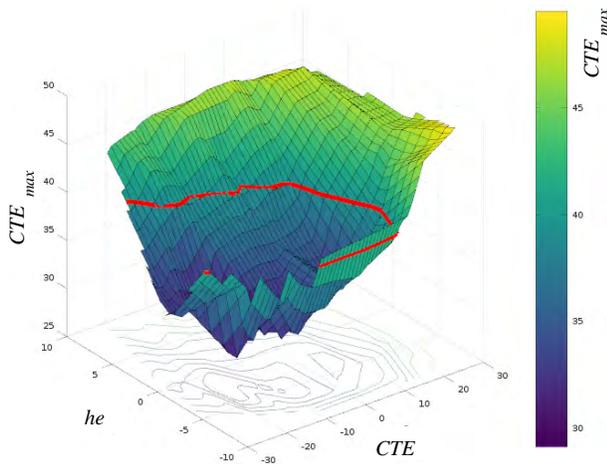


Figure 7. Safety-envelope: surface shows estimated maximal CTE_{max} value during a run over initial position CTE and heading of the aircraft he . The safety envelope at a given threshold of 40ft is shown as a red line.

Figure 7 shows how the safety envelope for ATC under different initial conditions CTE_0 and he_0 develop. For a small threshold, only runs with initial values close to $CTE_0 = 0$ and $he_0 = 0$ are successful, i.e., that our safety conditions is never violated during a run. This safety envelope becomes larger as the value for θ increases. The red line in Figure 7 shows its boundary for $\theta = 40ft$. SYSAI has been used to effectively create a model of this surface; a geometric characterization of the boundary can be obtained from SYSAI.

So far, all experiments were carried out in clear conditions

and 0900 local time. We then extended the experiment to include the time-of-the-day as an additional parameter. Obviously, ATC will not perform well in darkness, but it is important to know if ATC is performing differently at different times during the day.

Figure 8A shows the overall success rate in percent for a safety threshold of 40ft. The success rate varies tremendously during different times of the day and is only satisfactory between around 9AM and 1PM local time. Outside this time window, the performance of ATC is dropping sharply. A closer look at the images captured by the camera reveals the reason: Figure 8 shows typical images for a run at 9AM, 11AM, and 3PM, respectively. Compared to the 11AM run (middle panel), the early morning image is much darker. Since our version of ATC only has been trained with brighter images only, it is obvious that ATC performs not well in the earlier morning hours. The image on the right, taken during a 3PM run shows that the shadow of the aircraft is clearly visible and thus dramatically changing the overall image. Unless ATC had been trained on images like that, its performance is likely to be strongly diminished. Similarly, additional environmental parameters, like a wet runway, snow, or a cloudy sky can be modeled and analyzed with SYSAI.

The information obtained during the SYSAI analysis is then used to set up the R2U2 properties and monitors. We can distinguish between three different categories of R2U2 properties: (a) universal properties, (b) temporal properties, and (c) probabilistic properties. Universal properties are supposed to be valid throughout the entire operation and are necessary to define many safety properties. For example, $v_w < 5m/s \wedge 0 \leq v_w$ makes sure that the speed of the aircraft is always limited and that the aircraft never rolls backward. Within the R2U2 monitor, such properties are usually linked to conditions or system modes. In our example, this condition is only to be checked if the ATC system is on and the AC in taxi mode M^{AC} . This will yield:

$$\square((ATC_{on} \wedge M^{AC} = TAXI) \rightarrow (v_w < 5 \frac{m}{s} \wedge 0 \leq v_w)) \quad (2)$$

Temporal R2U2 properties can be used to specify

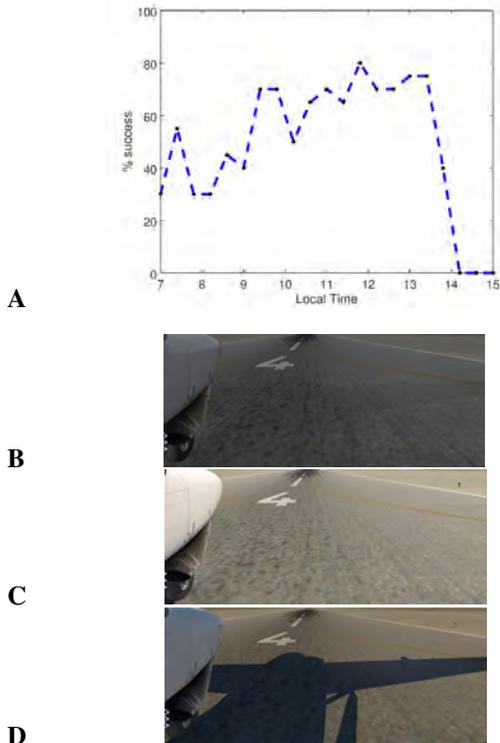


Figure 8. A: Success rate (in %) for different times of the day. Threshold for *CTE* is 40ft. Camera images taken from runs at 8AM (A), 11AM (B), and 3PM (C).

- overall performance properties, e.g., the end of the runway should be reached within 4-5 minutes:

$$M^{AC} = \text{TAXI} \rightarrow \diamond_{[4,5min]}(d_{rwy} > 0.9 * L_{rwy}) \quad (3)$$

- filtering of transients. For example, the outputs of the DNN should always lie in a certain range, e.g., $he \in [-10, 10]$. However, transients, yielding values outside that range should be tolerated if they are short enough, for example, less than 2 seconds:

$$(-10^\circ \leq he \leq 10^\circ) \vee \neg H_{[2s]}(he < -10^\circ \vee he > 10^\circ) \quad (4)$$

- limiting the number of occurrences of events. For example, it can be specified that no more than 3 transients occur within a period of 20 seconds. Such a property can also be seen as a discrete form of specifying error rates.

Such properties can be defined using the original signals (e.g., *cte*, *he*), or results of signal processing. In our case study, we use:

- signal rates, as approximation of signal derivatives are used to help monitor the system dynamics,
- sum or integration is used to check for biases over time,
- fast Fourier Transformation of signals are helpful in detection of oscillations. Such effects, similar to pilot-induced

oscillations can lead to dangerous situations that need to be avoided.

- Kalman filtering can be used for sensor fusion or to check the behavior of a signal against a given dynamical model. In this case study, Kalman filters have not been used.
- prognostics algorithms can be used to estimate the state of important components, e.g., the battery in electrical AC. R2U2 can, for example, check that there is always enough battery to taxi down the full runway. (not used in this case study)

For our case study, we used the signals from ACT and the aircraft as shown in Table 1. A more realistic case study would include numerous additional signals and sensor outputs (e.g., GPS, runway maps, etc).

Table 1. Signals used for R2U2 in the ACT case study. The column *S* indicates if signal processing is used. Signals in the lower part are used for failure monitoring (see below)

Name	S	Description
<i>cte</i>	•	DNN output cross-track error
<i>he</i>	•	DNN output heading error
<i>v_w</i>	•	AC front wheel speed
<i>d_{rwy}</i>	•	distance on runway (est)
<i>α</i>	•	steering angle
<i>ATC_{on}</i>		Boolean: ATC system on
<i>AC_{mode}</i>		AC mode (e.g., taxi, takeoff)
<i>T_{UTC}</i>		current on-board time
<i>I_{bright}</i>	•	image brightness
<i>I_{contr}</i>	•	image contrast
<i>I_{block}</i>	•	image blockage

5.3. Monitoring of Failure Conditions

As demonstrated above, it is important to monitor the behavior of the AI component in nominal operating conditions. Equally important, if not more important, however, is the monitoring of the complex AI function in case of failures. Failures can be the result of an unexpected environmental condition, e.g., fog or snow, or problems and faults with the sensors and actuators.

For traditional systems, fault detection and diagnosis systems are used to detect, isolate, and react upon the fault. In many cases, such systems are model-based and rely on the detailed knowledge about the system behavior in the failure case.

AI systems, on the other hand, are often considered black-box, i.e., they cannot explain or describe their behavior while in operation. This, well-known problem of explainability of AI and the fact that AI systems often have to operate in a huge, high-dimensional state space makes it impossible to perform coverage testing during V&V time.

In our architecture, we use information from SYSAI on failure analyses, to derive powerful runtime monitors that can be checked with R2U2.

5.3.1. Camera Failures

As a motivating example, let us consider a failure in the camera system: a piece of dirt or an insect on the lens is obstructing a part of the image. Image data in the obstructed region are consistently dark. Obviously, the ACT DNN has some robustness against such situations.

However, for improved system safety, we need to dynamically monitor, if the obstruction may lead to situations, where ACT fails. An analysis of this failure type revealed that the impact of an obstructing piece of dirt on the behavior of ACT is far from trivial: there are many regions of the image, where such an obstruction does not pose any restriction, which means that the DNN robustness is taking care of that situation.

However, SYSAI detected certain regions, where an obstruction can have notable and even severe consequences. Figure 9 shows a typical ACT camera image. The "dirt" is modeled in this case as a black square. Super-imposed on this image are the boundaries of the sensitivity regions as detected by SYSAI. Inside the green boundary, a considerable risk of ACT failure exists; inside the region defined by the red boundaries, a high risk is imminent.

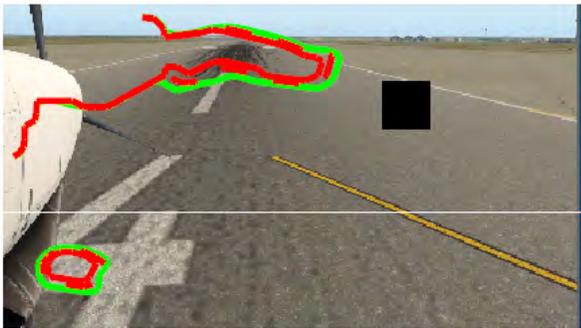


Figure 9. ACT camera image with "dirt" spot (black rectangle) and superimposed boundary lines for high risk regions

These regions obviously have to do with the way, the DNN has been trained to perceive the visual situation. As expected, areas near the horizon are of concern. The second, high risk area is close to the front wheel and it is suspected that the DNN uses this region to detect the runway center line or other optical markings. However, these regions are not obviously explainable and strongly depend on how the DNN has been trained.

With our R2U2 architecture, we are now able to use this information to produce an effective and powerful runtime monitor to detect such situations. By using SYSAI analysis data, we can avoid over-conservative monitors that would shut down the AI component as soon as a spot of a certain size is detected, causing numerous false alarms. Still the detailed SYSAI analysis provides the necessary confidence in the AI be-

havior that allows the R2U2 monitor to behave safely.

More specifically, our R2U2 monitor consists of the following R2U2 components and specifications

- a simple, traditional detection algorithm for camera obstructions. This algorithm returns a Boolean array, indicating the locations of obstructions.
- an R2U2 matcher that matches obstruction regions with a heat-map produced by SYSAI (with boundaries similar to Figure 9). This code is also traditional.
- an obstruction-risk value is calculated, using a weighted sum of the obstructions with the boundaries, and fed, after thresholding into the R2U2 temporal reasoner
- temporal formulas now check this signal, trying to weed out transient signals, checking persistence of the obstruction, and correlating with potential other failures or situations. E.g., taxiing after dark should not trigger the camera-obstruction monitor.
- the resulting signal is merged with results from the other R2U2 monitors to produce a final verdict to be sent to the RTA switch. In this case study, we are using Boolean conditions for that; a more elaborate monitoring variant would feed these monitoring results into a Bayesian network for probabilistic reasoning and calculation of confidence levels.

6. RELATED WORK

Runtime monitoring and runtime verification is mainly focusing on checking safety or security properties while the system is in operation (see e.g., (Havelund, Reger, & Rosu, 2019) for an overview). Violations usually cause alarms and can lead to drastic mitigation actions. Furthermore, most runtime monitoring systems are only concerned with model-based or (temporal) logic-based property checking. R2U2 also features efficient Bayesian reasoning, which seems to be a major help in the analysis of the, by nature, probabilistic DNNs.

In contrast to most related work, which aims at supporting property checking for V&V and safety purposes, we use R2U2 to switch between the AI component and different fall-back components in order to dynamically select a safe and suitable one. Our architecture is somewhat inspired by the ASTM3269 Runtime Assurance (RTA) architecture (ASTM, Nov 2021; Nagarajan et al., 2021), where a safety-monitor can switch from a complex, unassured component (e.g., a DNN) to some assured fall-back function, but aims to fulfill a different purpose.

7. CONCLUSIONS

In this paper, we have presented an advanced architecture to monitor the safety and performance of a complex AI component (e.g., a DNN) within an aerospace system. Inspired

by the ASTM RTA, we are using the R2U2 runtime monitoring system to dynamically check numerous properties, using temporal logic observers, Bayesian reasoners, and signal processing.

Our SYSAI statistical analysis framework can provide models, parameters, and other information to R2U2 to enable the definition of complex, yet justified properties that go ways beyond traditional range and rate checking monitors.

Future work will include the use of dynamic statistical reasoners and prognostic engines to extend this architecture into a fully statistical monitoring system, which can reason and decide with probabilities and confidence levels—a prerequisite for monitoring systems like Deep Neural Networks. We are also planning to work toward the use of this architecture and process in certification and risk management.

REFERENCES

- ASTM. (Nov 2021). *ASTM F3269 - 17 Standard Practice for Methods to Safely Bound Flight Behavior of Unmanned Aircraft Systems Containing Complex Functions*.
- Cohn, D. A. (1996). Neural network exploration using optimal experimental design. *Advances in Neural Information Processing Systems*, 6(9), 679–686.
- EASA. (2021). *Easa concept paper: First usable guidance for level 1 machine learning applications* (Tech. Rep.). European Aviation Safety Agency.
- Geist, J., Rozier, K. Y., & Schumann, J. (2014). Runtime Observer Pairs and Bayesian Network Reasoners On-board FPGAs: Flight-Certifiable System Health Management for Embedded Systems. In *Proceedings Runtime Verification (RV14)* (pp. 215–230). Springer.
- Gramacy, R., & Polson, N. (2011). Particle learning of Gaussian process models for sequential design and optimization. *Journal of Computational and Graphical Statistics*, 20(1), 467–478.
- Havelund, K., Reger, G., & Rosu, G. (2019). Runtime verification past experiences and future projections. In B. Steffen & G. J. Woeginger (Eds.), *Computing and software science - state of the art and perspectives* (Vol. 10000, pp. 532–562). Springer. doi: 10.1007/978-3-319-91908-9_25
- He, Y. (2012). *Variable-length functional output prediction and boundary detection for an adaptive flight control simulator* (Unpublished doctoral dissertation). University of California at Santa Cruz.
- He, Y. (2015). Online detection and modeling of safety boundaries for aerospace applications using active learning and bayesian statistics. In *2015 international joint conference on neural networks, IJCNN 2015, killarney, ireland, july 12-17, 2015* (pp. 1–8). IEEE. Retrieved from <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7256526> doi: 10.1109/IJCNN.2015.7280595
- He, Y., & Schumann, J. (2020). A framework for the analysis of deep neural networks in aerospace applications using bayesian statistics..
- He, Y., Yu, H., Brat, G., & Davies, M. (2021). Statistical learning framework for safety and failure analysis of a DNN-based autonomous aircraft system. IEEE.
- He, Y., Yu, H., Brat, G., & Davies, M. (2022). System and safety analysis for autonomous center line tracking with sysai..
- Jones, D., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black box functions. *Journal of Global Optimization*, 13, 455–492.
- MacKay, D. J. C. (1992). Information-based objective functions for active data selection. *Neural Computation*, 4(4), 589–603.
- Nagarajan, P., Kannan, S. K., Torens, C., Vukas, M. E., & Wilber, G. F. (2021). Astm f3269 - an industry standard on run time assurance for aircraft systems. In *Aiaa scitech 2021 forum*. doi: 10.2514/6.2021-0525
- Ranjan, P., Bingham, D., & Michailidis, G. (2008). Sequential experiment design for contour estimation from complex computer codes. *Technometrics*, 50(4), 527–541.
- Reinbacher, T., Rozier, K. Y., & Schumann, J. (2014). Temporal-Logic Based Runtime Observer Pairs for System Health Management of Real-Time Systems. In *Tools and Algorithms for the Construction and Analysis of Systems - 20th International Conference, TACAS (Vol. 8413, pp. 357–372)*. Springer.
- Rozier, K. Y., & Schumann, J. (2017). R2U2: tool overview. In *Proceedings rv-cubes 2017* (pp. 138–156).
- Schumann, J., Roychoudhury, I., & Kulkarni, C. (2015). Diagnostic reasoning using prognostic information for unmanned aerial systems. In *PHM15*.
- Schumann, J., Rozier, K. Y., Reinbacher, T., Mengshoel, O. J., Mbaya, T., & Ippolito, C. (2015). Towards Real-time, On-board, Hardware-supported Sensor and Software Health Management for Unmanned Aerial Systems. *International Journal of Prognostics and Health Management*.
- Taddy, M. A., Gramacy, R. B., & Polson, N. G. (2011). Dynamic trees for learning and design. *Journal of the American Statistical Association*, 106(493), 109–123.

Machine Learning Methods for Health-Index Prediction in Coating Chambers

Clemens Heistracher¹, Anahid Jalali², Jürgen Schneeweiss³, Klaudia Kovacs⁴, Catherine Laflamme⁵ and Bernhard Haslhofer⁶

^{1,2} *AIT Austrian Institute of Technology, Vienna, 1210, Austria*
Clemens.Heistracher@ait.ac.at
Anahid.Jalali@ait.ac.at

³ *D. Swarovski KG, Wattens, 6112, Austria*
Juergen.Schneeweiss@swarovski.com

^{4,5} *Fraunhofer Austria Research, Vienna, 1040, Austria*
catherine.laflamme@fraunhofer.at
klaudia.kovacs@fraunhofer.at

⁶ *Complexity Science Hub, Vienna, 1080, Austria*
haslhofer@csh.ac.at

ABSTRACT

Coating chambers create thin layers that improve the mechanical and optical surface properties in jewelry production using physical vapor deposition. In such a process, evaporated material condensates on the walls of such chambers and, over time, causes mechanical defects and unstable processes. As a result, manufacturers perform extensive maintenance procedures to reduce production loss. Current rule-based maintenance strategies neglect the impact of specific recipes and the actual condition of the vacuum chamber. Our overall goal is to predict the future condition of the coating chamber to allow cost and quality optimized maintenance of the equipment. This paper describes the derivation of a novel health indicator that serves as a step toward condition-based maintenance for coating chambers. We indirectly use gas emissions of the chamber's contamination to evaluate the machine's condition. Our approach relies on process data and does not require additional hardware installation. Further, we evaluated multiple machine learning algorithms for a condition-based forecast of the health indicator that also reflects production planning. Our results show that models based on decision trees are the most effective and outperform all three benchmarks, improving at least 0.22 in the mean average error. Our work paves the way for cost and quality optimized maintenance of coating applications.

Clemens Heistracher et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

Thin-film coatings can manipulate the optical properties of smooth surfaces and create color effects and unique reflection properties in jewelry. Metallic or dielectric layers create color effects and define the reflection coefficient. Such layers are created through the deposition of vaporized material in a vacuum chamber. These processes require high stability since these effects require layer thickness in the order of the wavelength of the optical spectrum. A typical layer thickness ranges from 100 nm to several micrometers. Machine operators perform regular maintenance and cleaning of the production equipment to guarantee process stability and product quality. The removal of deposits on the vacuum chamber's wall is the primary goal of these activities.

Maintenance operations are commonly scheduled based on the number of runs since the last procedure. However, the deposits on the walls and their impact on the process depend strongly on the materials and recipes used. Thus, process engineers expect significant cost savings from a maintenance schedule based on the actual condition of the vacuum chamber. An ideal maintenance schedule would perform all required operations to prevent failures while minimizing operations to save costs. However, such a predictive maintenance approach requires knowledge of the current and especially the future condition of the vacuum chamber, which is not known for our application. Therefore, we propose a novel method to estimate the health condition of the vacuum chamber.

Several studies have already focused on Health-Index assess-

ment for industrial assets in manufacturing (Khoddam, Sadeh, & Pourmohamadian, 2016), semiconductor productions (Djeziri, Ananou, Ouladsine, Pinaton, et al., 2015) and specially for vacuum equipment such as vacuum pumps (Jung, Zhang, & Winslett, 2017).

However, none of those approaches combines the Health-Index assessment with a forecasting model and uses it for maintenance optimization in coating chambers. We aim to predict the future health status of a coating chamber to optimize the scheduling of maintenance activities. To reach this goal, we performed an extensive exploratory data analysis and modeling and can summarize our contributions as follows:

1. We derived a Health-Index to describe the degradation in a coating chamber
2. We evaluate various shallow- and deep learning models for health index prediction on real production data

Our work provides insights into creating a Health-Index for vacuum chambers that require periodic maintenance and serves as a guideline in model selection for machine learning practitioners and process engineers. In the following, we briefly introduce related background on coatings with thin layers and the forecasting of a Health-Index (Section 2). Then in Section 3, we present our data set, exploratory data analysis, and the method of creating a Health-Index, before describing our experiments in Section 4 and our results in Section 4.

2. BACKGROUND

2.1. Decorative Coatings

Physical vapor deposition (PVD) is the method of choice to create color effects for decorative elements (Reiners, Beck, & Jehn, 1994). It produces thin layers on surfaces that form wave interferences with the reflected light, influencing the perceived color and reflection. These effects can be seen in soap bubbles or thin oil films on water. The demand for the layer's thickness precision is high as it has to be far below the typical optical wavelength to create a consistent color effect and requires extensive process control. The industrial production, multiple articles are treated in parallel in coating chambers. The primary working mechanism of such a chamber is the evaporation of substrate, which then condenses on the target. Due to their refraction coefficient, aluminum, zinc, and titanium are suitable choices (Jehn, 1992). Evaporation occurs in a vacuum to avoid collisions, oxidation, and surface contamination. Additional ion bombardment improves the mechanical properties of the surface. (Baptista, Silva, Porteiro, Míguez, & Pinto, 2018)

2.2. Data-Driven Predictive Maintenance Approaches

Through the advancement of technology and specifically since the introduction of industry 4.0 with the core concept of building smart factories, production, and logistics, the strategies

for Predictive Maintenance gained more popularity (Zhang, Yang, & Wang, 2019). PdM approaches can be grouped into three; model-based, knowledge-based, and data-driven (Zonta et al., 2020). We narrow the scope and only focus on state of the art for data-driven approaches. A group of these studies focuses on data pre-processing and more feature engineering approaches, combined with shallow and statistical modeling. For example, (Umeda, Tamaki, Sumiya, & Kamaji, 2021) proposed a maintenance schedule updater based on probabilistic variability of the Remaining Useful Life (RUL) and maintenance costs, which is component agnostic. (Chien & Chen, 2020) used Partial Least Square supervised learning to model the fault detection and classification parameters of glass substrates for Thin Film Transistor Liquid-Crystal Display (TFT-LCD) manufacturing process. Other approaches use complex modeling techniques, such as Deep Neural Networks (DNNs), to predict and analyze equipment failures. (Sateesh Babu, Zhao, & Li, 2016) were the first to use Convolutional Neural Networks for the RUL estimation in turbine engines. (Deutsch & He, 2018) showed that restricted Boltzmann machines could effectively predict RUL in gears. (Liu, Zhao, & Peng, 2019) studied the RUL estimation for lithium-ion batteries using long short-term memory-based neural networks. (Huuhtanen & Jung, 2018) build a neural network model for predictive maintenance of photovoltaic panels.

Predicting a health indicator (HI) is a related research area. Statistical analysis was performed to derive a HI for power transformers (Murugan & Ramasamy, 2019), transmission lines (Thongchai, Pao-La-Or, & Kulworawanichpong, 2013), bearings (Pan, Chen, & Guo, 2009), electrical machines (Yang et al., 2016), semiconductor production (Chen & Blue, 2009) and bridges (Döhler, Hille, Mevel, & Rücker, 2014).

3. METHOD

Our work aims to provide a guideline for implementing a predictive maintenance system for coating chambers in jewelry production. This section presents the details of the real-world dataset and outlines all significant steps in creating a health index and a time-series prediction model.

3.1. Dataset

We use data from the real-world production of decorative elements obtained in 15 months. The dataset consists of sensor recordings and process parameters of five assets for the deposition of optical layers. The sensor data consists of recordings directly from the process chamber, which are used to control the conditions in the chamber during production. It contains temperature, pressure, gas flow recordings, and the electrical parameters of evaporators and ion sputter components. Typically, the process parameters are the settings of equipment that describe the desired conditions, such as the active duration of an ion source or the target pressure. We will refer to

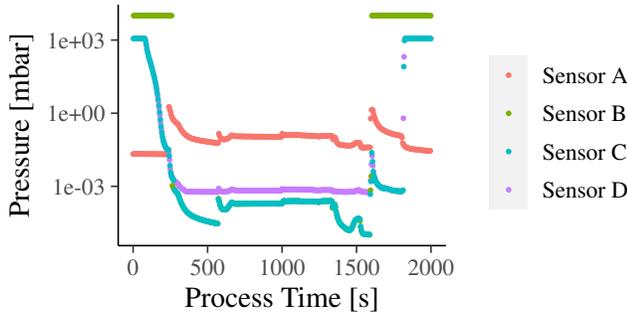


Figure 1. Pressure curves of a single run recorded on four sensors on a logarithmic scale.

these sets of rules as recipes since they are distinct for a type of product and color effect. Additionally, the dataset contains maintenance-related information, such as the number of runs since the last cleaning. The vacuum system consists of three parts: The initial stage is a backing pump with an operating range at atmospheric pressure. Then, a turbomolecular pump starts at vacuum pressures, and a cold trap is activated. These stages are monitored and controlled by four pressure sensors in the vacuum chamber due to the high range of pressures from 10^3 mbar to 10^{-4} mbar and the limited measuring range for single sensors. Figure 1 shows the available pressure curves for a single production run.

Our dataset consists of 2137 production runs of five assets containing sensor measurements for every 0.5 seconds. In addition, we recorded 119 numerical columns and 13 categorical columns containing product and recipe information and timestamps. Thus, the whole dataset contains approximately 10 million samples of 134 features.

3.2. Health-Index Derivation

Health indicators describe equipment conditions from a maintenance perspective and are used to implement adequate repair and service activities. A typical health index for coating chambers is the deposit accumulation in the vacuum chamber (Li et al., 2019), which can be measured by piezo-based sensors (Benes, Gröschl, Burger, & Schmid, 1995). However, this approach only measures the layer thickness and does not consider the potential interaction of different layer materials, recipes, and unknown effects. Domain experts assume that layers of alternating materials impact the stability of the deposit and its ability to outgas and thus the condition of the chamber. We believe that the effect of alternating materials can be learned by models when provided with recipe information and the corresponding sensor data. Thus, our goal is to consider different materials' impact and potential interaction, which requires further investigation, and therefore, we

require a HI derived from the current measurements.

We based the development of our health indicator on domain experts' observation that the pumping duration correlates with the condition of the chamber. Additionally, (Field, Bellum, & Kletecka, 2016) supports this assumption and shows a correlation between pumping duration and the quality of produced coatings for laser applications. Before and after every production run, operators open the vacuum chamber to load or unload the products, and air at atmospheric pressure fills the chamber while the door is open. Multi-stage vacuum pumps evacuate the chamber before every coating procedure, and domain experts have noticed that this pumping takes longer the more contaminated the chamber is.

We assume that the pumping duration corresponds to the condition of the vacuum chamber and apply a variety of models and visual analyses to validate this assumption. First, we selected an asset and time frame that uses a single standard recipe to rule out recipe-dependent factors. Our subset for this analysis consists of the pressure curves from four sensors for 400 runs and their corresponding maintenance information. Then, we identified several pressure intervals ($\Delta p_1, \dots, \Delta p_n$) based on actual steps in the production process. For instance, the first interval starts at atmospheric pressure, which is the condition at the beginning of the process, and ends at 0.02 mbar, the pressure at which the turbopump is activated. We further extract the processing time of all pressure intervals and create variables T_i with $i \in [1, 2, 3, 4]$ that correspond to the time it took the pumps to evaluate each interval. For instance, T_1 is the process duration from atmospheric pressure to 0.02 mbar. Then, we merged the data with the maintenance information n_{runs} , which is the number of runs since the last maintenance procedure. Further, we built models to understand the correlation between maintenance conditions and pumping time. In detail, we fit a linear regression model to the data of each pressure range by using the number of runs as input and the pumping time as the target variable. The regression is of the form:

$$T_i = k_i * n_{runs} + d_i \quad (1)$$

k_i is the slope of the regression in $[\text{seconds/run}]$ for the i th segment, and d_i is an additive constant. We defined an impact variable that indicates the relative change of the pumping duration over a complete cleaning cycle consisting of 100 runs. The impact variable is defined as $\alpha_i = k_i / \bar{t}_{pump_i} * 100$, with \bar{t}_{pump_i} being the average pumping duration for a clean machine, which is defined as the mean pumping duration for the first 10 runs after cleaning for the i th segment over all cycles. We quantified the fit based on the coefficient of determination, which roughly gives the proportion of variance that can be explained by the model and the impact variable.

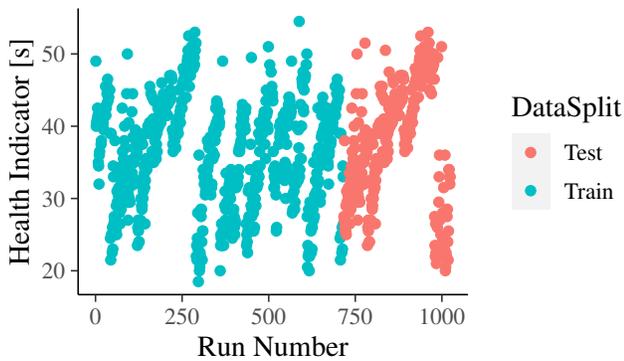


Figure 2. The target variable for a period of 12 months. The colour indicates the train set (blue) and test set (red)

3.3. Health-Index Forecasting

Maintenance planners require estimates of the future evolution of an asset to provide adequate resources when needed. This task aims to forecast the future contamination of coating chambers. Our approach utilizes the health index (HI) we derived in section 3.2, which measures the contamination of a vacuum chamber based on the time it takes to evacuate. We predict the future HI based on the process data and knowledge of the planned recipes available at the current run, which can be seen as a regression problem. We predict the value of the HI ten runs after the current run, which roughly corresponds to a day of production and is the typical time available for production and maintenance planning in our application. We use the recipes in the subsequent ten runs in ten one-hot encoded features. Each feature characterizes the recipe used in a future run. The HI over the period of a year is illustrated in Figure 2.

We aggregate the process data by calculating each run’s mean, minimum, maximum, and standard deviation and use one-hot encoding to encode categorical features such as recipes. We split the dataset into a train and test set to evaluate our model on an independent dataset. We train with the oldest 70% of the data and test with the most recent 30%. We selected a range of machine learning models that were used in related tasks such as Decision Trees (DT), Random Forest (RF), K-Nearest Neighbors (KNN), Multilayer Perceptrons (MLP), and Recurrent Neural Networks with Long Short-Term Memory units (LSTMs).

We evaluate our models using the mean average error (MAE) calculated on the prediction and the actual value of the HI. We design benchmarks based on naive assumptions to put our predictions into perspective. Benchmark 1 (BM1) uses the current values as the prediction. We calculate the average curve of a cleaning cycle in the train set and use it as benchmark 2 (BM2). Benchmark 3 (BM3) uses the average of all data points in the train set.

Table 1. Regression of pumping duration and chamber contamination

i	Segment	k_i [s/run]	\bar{t}_{pump} [s]	α_i [1/cycle]	$R2$
1	$\Delta p1$	0.06	139	5%	0.19
2	$\Delta p2$	0.12	21	55%	0.61
3	$\Delta p3$	0.45	285	28%	0.10
4	$\Delta p4$	0.85	305	17 %	0.11
5	$\Delta p5$	0.19	160	12 %	0.39

4. RESULTS

In the following, we present our results for the derivation of a HI for coating chambers and a forecasting model to predict the future condition of the chamber.

4.1. Health-Index Derivation

The main goal of this task is to derive a HI from the process data, which describes the condition of the coating chamber. We observe that the overall process duration increases with the contamination of the vacuum chamber. We use the pumping duration for various pressure steps and model the maintenance data to determine the impact on the production of each step. Table 1 shows the coefficient of determination ($R2$) of our models, and the impact variable α_i . $R2$ indicates how much of the variation can be explained by our model. The pumping duration for segment $\Delta p2$ has the highest correlation with contamination at an $R2$ score of 0.61 and an increase of 55% over one cleaning cycle. Therefore, we propose the pumping duration of $\Delta p2$ as a new health indicator for coating chambers and will refer to it as T2.

In Figure 3, we illustrate a high number of pressure curves for the segment $\Delta p2$. We align the pressure curves when they reach the upper limit of $\Delta p2$ at 0.03 mbar, and clip the curves at the lower limit at 0.002 mbar. Different colors indicate the maintenance conditions, which are the number of runs since the last cleaning cycle. We observe that the pressure curves decrease with the contamination of the chamber.

Our results show that the contamination of the chamber can be determined by the pressure curves of the production process, and therefore, the pumping duration is a suitable HI for a coating chamber.

4.2. Health-Index Prediction

The main goal of this task is to predict the future health indicator based on information available at the current run. We evaluate the well-established machine learning models such as Support Vector Machines (SVM) (Wan, McLoone, English, O’Hara, & Johnston, 2014), Decision Trees (DT), and Random Forests (RF) (Scheibelhofer, Gleispach, Hayderer, & Stadlober, 2012), and Neural Network architectures optimized for time series (Bruneo & De Vita, 2019). We evaluate our models on an independent dataset, which consists of the

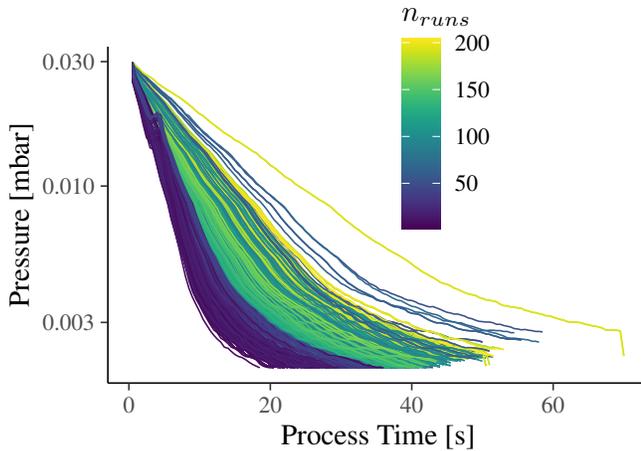


Figure 3. The aligned pressure curves for the segment Δp_2 . n_{runs} is indicated by the colour.

Table 2. Results forecast health indicator.

model	MAE	BM2	BM1	BM3
SVC	3.26	9.0	3.22	8.37
DT	3.00	9.0	3.22	8.37
RF	3.28	9.0	3.22	8.37
KNN	3.81	9.0	3.22	8.37
MLP	5.56	9.0	3.22	8.37
LSTM	10.03	9.0	3.22	8.37

most recent 30% of data, and compared them with the previously defined benchmarks.

Our results, summarized in Table 2, indicate that shallow machine learning models significantly outperform neural network-based approaches and, therefore, are better suited for this task. The DT achieved the best mean average error (MAE) with 3.0, followed by SVM with a score of 3.26 and RF with 3.28, outperforming all three benchmarks. However, the naive prediction of BM1 scored 3.22 and is only slightly outperformed by the best model and outperformed all others.

We illustrate the results of our best models for several cleaning cycles in Figure 4. The shape of the target variable resembles a sawtooth function with a linear ascend during production and a sudden drop after maintenance, which the model appears to have learned. Benchmarks 2 and 3, learned from the train set, are shifted towards lower values, which explains the poor performance of these benchmarks. Seasonal patterns in temperature and humidity are explanations provided by domain experts. Since the available data spans one year only, we cannot confirm or reject this thesis.

5. DISCUSSION

A goal of cost-effective maintenance planning is to optimize the timing and type of scheduled activities. This requires

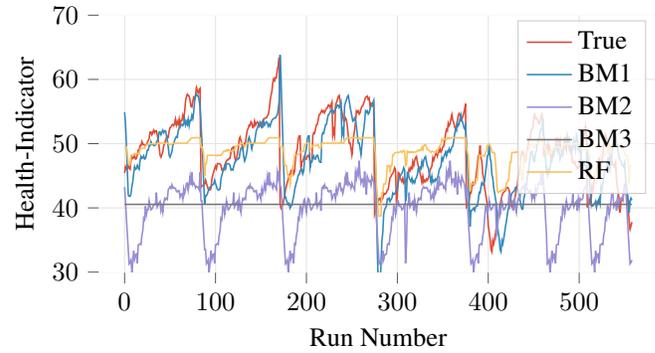


Figure 4. The target variable, prediction and benchmarks of our HI.

knowledge of the current and future condition of the asset, which is often not available. Our work aims to provide this information through data analytics and machine learning models using process data. This paper presents the derivation of a health indicator that corresponds to the contamination of a coating chamber and a forecasting model to predict the future health indicator based on current process data and planned recipes. We based the health indicator development on the domain experts' observation that the time to evacuate the vacuum chamber correlates with its contamination. The adhesion of water on the contaminated walls that evaporates at low pressures is a possible explanation for this behavior. We analyzed the impact of the contamination on the pumping time for various pressure ranges and identified a pressure region that strongly correlates with the cleaning cycle. Our results indicate that this pumping duration could serve as a health indicator for coating chambers. Further, we developed predictive models that take the current condition of the chamber and the planned recipes into account and give a prognosis on the future development of the health indicator.

One limitation of our work is the connection between the chamber's performance and our health indicator. We showed that our health indicator correlates to the contamination of the chamber, but we did not provide evidence that it impacts the product quality or process stability. Although we believe it is plausible that outgassing in the vacuum chamber has a negative impact on production, this is an assumption that needs to be verified. (Ito et al., 2008) showed that dehydration of deposits reduced the number of defects for plasma etching equipment, and we believe that this is transferrable to coating chambers. We compared multiple machine learning models to predict our health indicator's future development and found that neural network-based architectures performed poorly compared to traditional machine learning approaches. However, the size of our data set could limit the full potential of neural networks since it is small compared to typical applications of neural networks, and we believe that a larger data

set could improve their performance.

Subsequentially, we can identify two challenges to a profound HI forecasting model. First, the actual impact of our health indicator on product quality must be evaluated. Our dataset originates from a preventive maintenance regime that averts most defects and therefore does not allow us to confirm the benefit of a prolonged maintenance cycle using our approach. Therefore, we need to lift the restrictions on the number of runs and evaluate our approach during actual production with the potential risk of creating defective parts. Second, the data set must be extended to include more assets of various designs and more data points. Finally, only results for multiple chamber types on a high number of samples will allow for results that show that our approach is generally valid.

6. CONCLUSION

The lack of methods to assess the actual condition of an industrial asset hinders the industry from moving away from regular maintenance to a more proactive approach. This paper followed the idea that already available process data contains enough information to derive a health indicator, and no additional hardware is required. We address the need for predictive maintenance approaches that are easy to deploy and can be scaled up quickly to many assets and gained some fundamental insight into the current possibilities and most promising future directions. In the coming months, we will evaluate the impact of our approach on the actual product quality during production.

ACKNOWLEDGMENT

This work was partly funded by the Austrian Research Promotion Agency (FFG) through the project COGNITUS (Project ID: 3323904).

REFERENCES

- Baptista, A., Silva, F., Porteiro, J., Míguez, J., & Pinto, G. (2018). Sputtering physical vapour deposition (pvd) coatings: A critical review on process improvement and market trend demands. *Coatings*, 8(11).
- Benes, E., Gröschl, M., Burger, W., & Schmid, M. (1995). Sensors based on piezoelectric resonators. *Sensors and Actuators A: Physical*, 48(1), 1-21.
- Bruneo, D., & De Vita, F. (2019). On the use of lstm networks for predictive maintenance in smart industries. In *2019 IEEE International Conference on Smart Computing (SmartComp)* (pp. 241–248).
- Chen, A., & Blue, J. (2009). Recipe-independent indicator for tool health diagnosis and predictive maintenance. *IEEE Transactions on Semiconductor Manufacturing*, 22(4), 522-535.
- Chien, C.-F., & Chen, C.-C. (2020). Data-driven framework for tool health monitoring and maintenance strategy for smart manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 33(4), 644–652.
- Deutsch, J., & He, D. (2018). Using deep learning-based approach to predict remaining useful life of rotating components. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(1), 11-20.
- Djeziri, M. A., Ananou, B., Ouladsine, M., Pinaton, J., et al. (2015). Health index extraction methods for batch processes in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 28(3), 306–317.
- Döhler, M., Hille, F., Mevel, L., & Rücker, W. (2014). Structural health monitoring with statistical methods during progressive damage test of s101 bridge. *Engineering Structures*, 69, 183–193.
- Field, E. S., Bellum, J. C., & Kletecka, D. E. (2016). How reduced vacuum pumping capability in a coating chamber affects the laser damage resistance of hfo 2/sio 2 antireflection and high-reflection coatings. *Optical Engineering*, 56(1), 011005.
- Huhtanen, T., & Jung, A. (2018). Predictive maintenance of photovoltaic panels via deep learning. In *2018 IEEE Data Science Workshop (DSW)* (p. 66-70).
- Ito, N., Moriya, T., Uesugi, F., Matsumoto, M., Liu, S., & Kitayama, Y. (2008). Reduction of particle contamination in plasma-etching equipment by dehydration of chamber wall. *Japanese Journal of Applied Physics*, 47(5R), 3630.
- Jehn, H. (1992). Decorative coatings. In *Advanced techniques for surface engineering* (pp. 359–370). Springer.
- Jung, D., Zhang, Z., & Winslett, M. (2017). Vibration analysis for iot enabled predictive maintenance. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)* (p. 1271-1282).
- Khoddam, M., Sadeh, J., & Pourmohamadiyan, P. (2016). Performance evaluation of circuit breaker electrical contact based on dynamic resistance signature and using health index. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 6(10), 1505–1512.
- Li, P., Jia, X., Sumiya, M., Kamaji, Y., Ishiguro, M., Pahren, L., & Lee, J. (2019). A novel method for deposit accumulation assessment in dry etching chamber. *IEEE Transactions on Semiconductor Manufacturing*, 32(2), 183-189.
- Liu, Y., Zhao, G., & Peng, X. (2019). Deep learning prognostics for lithium-ion battery based on ensemble long short-term memory networks. *IEEE Access*, 7, 155130-155142.
- Murugan, R., & Ramasamy, R. (2019). Understanding the power transformer component failures for health index-based maintenance planning in electric utilities. *Engi-*

- neering Failure Analysis*, 96, 274-288.
- Pan, Y., Chen, J., & Guo, L. (2009). Robust bearing performance degradation assessment method based on improved wavelet packet–support vector data description. *Mechanical Systems and Signal Processing*, 23(3), 669-681.
- Reiners, G., Beck, U., & Jehn, H. A. (1994). Decorative optical coatings. *Thin Solid Films*, 253(1), 33-40.
- Sateesh Babu, G., Zhao, P., & Li, X.-L. (2016). Deep convolutional neural network based regression approach for estimation of remaining useful life. In *International conference on database systems for advanced applications* (pp. 214–228).
- Scheibelhofer, P., Gleispach, D., Hayderer, G., & Stadlober, E. (2012). A methodology for predictive maintenance in semiconductor manufacturing. *Austrian Journal of Statistics*, 41(3), 161–173.
- Thongchai, P., Pao-La-Or, P., & Kulworawanichpong, T. (2013). Condition-based health index for overhead transmission line maintenance. In *2013 10th international conference on electrical engineering/electronics, computer, telecommunications and information technology* (p. 1-4).
- Umeda, S., Tamaki, K., Sumiya, M., & Kamaji, Y. (2021). Planned maintenance schedule update method for predictive maintenance of semiconductor plasma etcher. *IEEE Transactions on Semiconductor Manufacturing*, 34(3), 296–300.
- Wan, J., McLoone, S., English, P., O’Hara, P., & Johnston, A. (2014). Predictive maintenance for improved sustainability—an ion beam etch endpoint detection system use case. In *Intelligent computing in smart grid and electrical vehicles* (pp. 147–156). Springer.
- Yang, F., Habibullah, M. S., Zhang, T., Xu, Z., Lim, P., & Nadarajan, S. (2016). Health index-based prognostics for remaining useful life predictions in electrical machines. *IEEE Transactions on Industrial Electronics*, 63(4), 2633-2644.
- Zhang, W., Yang, D., & Wang, H. (2019). Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal*, 13(3), 2213–2227.
- Zonta, T., da Costa, C. A., da Rosa Righi, R., de Lima, M. J., da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, 150, 106889.

Approximate Bayesian Computation for the Analysis of Partial Discharge Data

Kai Hencken¹, Daniele Ceccarelli^{1,2}, Elsi-Mari Borrelli^{1,3} and Andrej Krivda¹

¹ *ABB Corporate Research Center, Baden-Dättwil, 5405, Switzerland*
kai.hencken@ch.abb.com, andrej.krivda@ch.abb.com

² *current address: IRCCS San Raffaele Scientific Institute, Milan, 20127, Italy*
ceccarelli.daniele@hsr.it

³ *current address: Algorithmiq Ltd, Kanavakatu 3C 00160 Helsinki, Finland*
elsi@algorithmiq.fi

ABSTRACT

Partial Discharges are short breakdowns inside electrical equipment. As they indicate weaknesses of the insulation strength, they are seen as important precursors to a failure of the system. Therefore measurement and analysis of the patterns of instances in time and strength of the discharge are an important tool to analyze the insulation status of electric equipment, that has been addressed already using different methods in the past. In this work we explore how a physics-based stochastic process can be combined with Approximate Bayesian Computation (ABC) as a new way to analyze them. ABC is a method to infer probability distributions of model parameters in cases, where the likelihood is not tractable, but simulations can be done easily. As such it is of interest for complex phenomena or measurement systems, as often found in prognostics applications. Especially the ABC-SMC method was found to be useful here. Real Partial Discharge measurement data was used not only for parameter estimation, but also to do model comparison in order to compare different physical models proposed in the literature.

NOMENCLATURE

$c(t)$	discharge rate above the inception voltage
c_0	constant discharge initialization rate
c_1	time dependent discharge initialization rate
f	line frequency
$M_{i,j}$	i, j th moment of the PRPD pattern rate distribution

Q_k	strength, apparent charge of the discharge event k (pC)
$Q(\theta, \theta')$	kernel function in the ABC-SMC algorithm
t_k	time of the discharge event k
$U_{ext}(t)$	line voltage
U_{inc}	inception voltage of the discharge
$U_{int}(t)$	voltage corresponding to the internal field within the discharge
U_{res}	residual voltage of the discharge
x	measurement data consisting of x_k $k = 1, \dots, K$
x_k	individual discharge consisting of charge Q_k and phase ϕ_k
γ	proportionality factor between voltage drop and discharge strength
ϵ	accuracy value for the ABC algorithm
ϕ_k	phase of the line voltage at the discharge event k
ν	rate for the reduction of the internal voltage
θ	summary of all discharge model parameters
τ	decay time of the discharge rate

1. INTRODUCTION TO SIMULATION BASED INFERENCE AND APPROXIMATE BAYESIAN COMPUTATION

“Approximate Bayesian Computation” (ABC) is the most common method used in the field of “likelihood-free” or “simulation-based” inference. These methods are used, if the statistical model under investigation is easy to simulate from, but the likelihood function, which is at the core of Bayesian or frequentist maximum-likelihood based inference, is not tractable. This means, that it is either not easily accessible to formulate or its numerical evaluation is computationally too demanding.

Kai Hencken et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ABC has gained popularity starting from its inception in the 1990s (Tavaré, Balding, Griffiths, & Donnelly, 1997) in a number of fields ranging from genetics, neural sciences, cosmology and particle physics (Cranmer, Brehmer, & Louppe, 2020). Whereas the underlying principle of ABC stays always the same, a large number of variations of algorithms have been developed in order to address different difficulties or to improve the numerical efficiency (Beaumont, 2019; Sisson, Fan, & Beaumont, 2019). Readily available software packages exist (Csilléry, François, & Blum, 2012; Dutta et al., 2021), simplifying the application, but given the rather easy implementation of Monte Carlo methods in general, a dedicated code was developed in this work.

The main aim of simulation-based or likelihood-free methods is to make use of the possibility to sample a large number of outcomes from the model in order to construct an approximate likelihood or posterior distribution. In the case of ABC one samples both the parameters θ and the measurements x according to the combined probability distribution $(x_n, \theta_n) \sim p(x, \theta)$. This combined distribution is given by $p(x|\theta)p(\theta)$, where $p(\theta)$ is the prior, capturing the knowledge about the parameters and $p(x|\theta)$ the intractable likelihood function underlying the process generating the measured data. By conditioning the samples (x_n, θ_n) on those measurement values x_n that are close to the observed value x , one is able to get samples θ_n from the posterior $\sim p(\theta|x)$. These samples are the basis for further steps in the analysis.

2. ADVANTAGE OF THE USE OF LIKELIHOOD-FREE INFERENCE FOR DIAGNOSTICS AND PROGNOSTICS APPLICATIONS

In most applications of prognostics and health management one investigates the possibility to assess the status of technical equipment. In this cases one has the advantage of a good understanding of its functional principle. This is the case, because they are man-made devices, built in order to fulfill a certain function. They are therefore often accompanied by a deep physical or technical understanding. Simulation models are created in many cases as part of the development process, or to analyze the underlying working principle. Such models — including the change of model parameters introducing or leading to faults — are of a high value for good diagnostics or prognostics approaches. Incorporating simulations as part of the algorithms is therefore a good way to capture this knowledge.

In the case of a deterministic simulation model and assuming error-free measurements, the determination of model parameters from the observations or measurements is an inversion problem. In general, this inversion will be ill-posed, meaning that it is numerically unstable to do so. In addition, measurements are in general noisy and even small errors in the measurements can lead to very different and even wrong pa-

rameter estimations.

If one is not only interested in a diagnostics, but also a prognostics approach, then for the calculation of the probability of failure (PoF(t)), the determination of the uncertainty of the state of the system, as well as those parameters defining the future dynamics, are required. The Bayesian inference, which is underlying the ABC approach, allows to do so in a consistent way.

The situation becomes even more complex, if the model is of a stochastic nature. Reasons for this, apart from the measurement errors, are often the presence of unknown values or the randomness inherent in some process.

The intractability of the likelihood is in general due to the existence of a large number of hidden, that is unobserved or unknown, parameters, states or values. Within the Bayesian approach one needs to marginalize over them, that is integrate over all possible values, which is numerically impossible in practice. These hidden variables can have different origins: Technical models contain a large number of parameters, which are varying from device to device, but are often not relevant for the degradation state of the system. They will nevertheless influence the way the device is operating and the values of measurements.

Other hidden variables are unmeasured external influences. These can be environmental or operational factors. As before their value will in general not be important for the state of degradation, but they are influencing the measured quantities.

Finally the measurement principle can have unobserved internal states. An example could be their dependency on earlier measurements, or a probabilistic element in the underlying process. Partial Discharge analysis can be seen as a problem of this type.

All these facts make simulation-based inference methods of interest for diagnostics or prognostics approaches. This is not restricted to the ABC method, discussed here. Methods combining simulations with elements of machine learning are gaining popularity in other fields and are of potential interest in this area as well. This can be interesting, if e.g. only a single remaining useful life (RUL) value should be predicted instead of a full distribution or the simulations need to be accelerated with the help of a surrogate model.

3. PARTIAL DISCHARGE MEASUREMENT TO ASSESS THE ELECTRIC INSULATION STATUS

Electrical equipment — especially in the high voltage area above few kV — is subject to strong electric fields, that are applied over a long time. Insulation material is known to be able to withstand these fields only up to a certain level and changes of material properties with time can lead to a reduction of this critical field strength. Typical examples are mate-

rial aging, changes due to exposure to water, corrosive gases, dust built-up, crack formation or forming of other defects due to mechanical or thermal stress.

The loss of the electrical insulation capability will ultimately lead to a complete failure of the equipment. This can, in connection with the appearance of electrical arcs, lead to fire or explosions and therefore to the complete destruction of it, resulting also in potential hazards to people. It is therefore of interest to be able to detect the reduction of the insulation strength of a system under voltage and its potential evolution with time.

“Partial Discharge” (PD) describes the phenomena, where the insulation strength is insufficient in only a localized region. The corresponding electrical breakdown will therefore be local only, not bridging the full distance from high voltage to ground. This “partial” electric breakdown is in general extinguished after a short time as the charges flowing during the discharge are reducing the local electrical field to a value below the critical one. The effect of this local field reduction will become weaker with time and due to the change of the externally generated electric field in AC applications, a new discharge will occur after some time.

PD is a rather generic term, which is given to any local electric breakdown. But these are often occurring in different parts of the insulation system, as well as due to different origins: At sharp metallic edges the local field enhancement leads to “corona” discharges; at surfaces the buildup of dust and humidity forms a conductive layer and leads to “surface discharges”; defects inside insulators, often introduced during production, will in general have a lower insulation strength and lead to the formation of “void discharges”. Starting from these initial defects, they will further erode the material leading to “treeing” or “tracking” inside or on the surface of the material, which develops and makes the defect worse over time.

According to statistical data, up to 85% of all severe failures of high-voltage equipment can be linked to the presence of partial discharges in those systems. They are one of the main precursors or indicators of the upcoming failure of the electrical insulation capability of a high-voltage system. Using monitoring systems to detect them is one of the most often used methods in high-voltage systems. Both dedicated testing and measurement systems manufacturer, as well as, producers of high-voltage equipment provide a variety of laboratory, off-line and online solutions.

PD measurements are done to assess the quality of individual parts or full system during their production and also as part of the acceptance testing, with limits of the allowed PD activity defined. Whereas this allows to capture defects, that are already present during production or installation, the degradation of the electrical insulation can only be tracked with an

online monitoring system.

An assessment of the insulation status is often done using external equipment and at fixed time intervals. Such measurements have the advantage of allowing for expensive, but very accurate equipment to be used, as well as providing some detailed diagnostics results, e.g. by changing the test voltage applied. On the other hand they require an expensive shutdown of the installation and only give a snapshot of the status at that specific time. Online monitoring systems on the other hand provide a continuous measurement, allowing for an early detection, but can also study the evolution with time. But they also need to be rather inexpensive in order to be deployed widely.

The evolution of the partial discharge with time is often summarized by a few key parameters, e.g. a strength and a discharge rate. But an estimate of the time until failure of the electrical insulation cannot be done easily from this. The type of discharge plays a rather important role: whereas a corona discharge can be present in a system for a longer time before a fault occurs, a treeing or surface discharge will often evolve to a full breakdown very fast.

The character of defects and with this the appearance of the discharge will change over time. This has been investigated as a way to infer the remaining lifetime until breakdown for specific defect types in a number of publications, see for example (Montanari, 1995; Wang, Cavallini, Montanari, & Testa, 2010; Lv, Rowland, Chen, Zheng, & Iddrissu, 2017).

In practical applications PD measurement will be disturbed by a number of external phenomena, which need to be distinguished from real PD events. Whereas this can be done in a lab setup with the help of shielding measures, this is not possible in online applications. A major goal of any online PD analysis is therefore to detect them, separate them from other disturbances, but also to classify them according to their origin, the type of discharge or defect, to get further information regarding their severity. Given their importance to avoid severe failures of those expensive systems, a number of detailed analysis approaches have been explored.

4. BASIC PRINCIPLE OF PARTIAL DISCHARGE MEASUREMENT AND ANALYSIS

The local discharge is in general characterized by the time t_k of occurrence and its strength, which is in general measured as “apparent charge” Q_t (International Electrotechnical Commission (IEC), 2000). Due to the stochastic nature of the discharge, the individual events (t_k, Q_k) don’t happen at deterministic times or repeat in a systematic way, see Fig. 1.

A full measurement therefore consists of a number of such discharge events $k = 1, \dots, N$, that are either recorded continuously or sometimes with gaps within them.

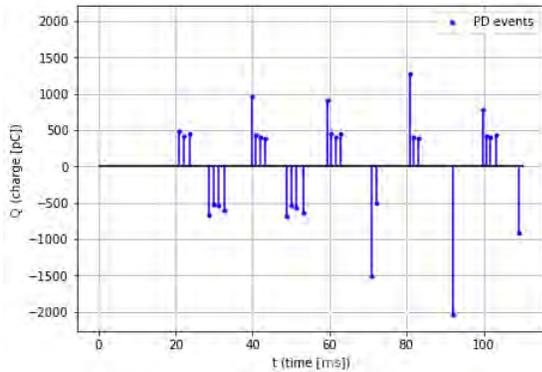


Figure 1. Partial Discharges occur stochastically at times t_k and with varying strengths Q_k . A typical sequence is shown schematically in this plot. A PD measurement consists of capturing a full sequence of such individual discharge events.

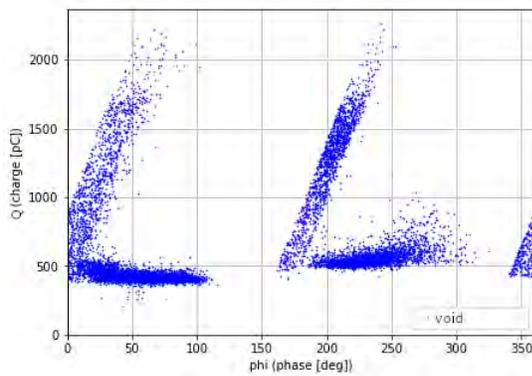


Figure 2. A typical PRPD pattern of a void discharge. The points are individual discharge events (ϕ_k, Q_k) of the phase and discharge strength.

As the AC voltage $U_{ext}(t)$ plays a dominant role in reaching the critical electric field level of the defect, its value at the moment of the discharge $U_k = U_{ext}(t_k)$ or alternatively the phase $\phi_k = \phi(t_k)$ are recorded as well. A number of methods have been proposed to analyze this sequence of individual PD events. The most commonly used one is the “Phase Resolved Partial Discharge” (PRPD) analysis. In this, one plots the points (ϕ_k, Q_k) in a two-dimensional scatter plot, or converting them into a density plot. As the partial discharge is driven by the applied AC line voltage, one expects that the pattern formed in these graphs reflects the nature of the discharge. In addition, depending on the measurement approach used, one is not able to measure the polarity of the discharge Q_k . Therefore we will in the following assume that only the absolute value of the discharge is available as Q_k . A typical PRPD plot is shown in Figure 2 of a void, and in Figure 3 of a corona discharge.

Different PD types, but also different external disturbances,

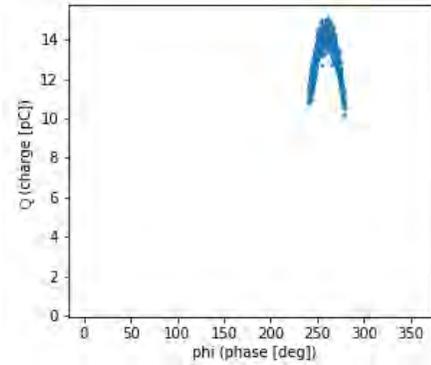


Figure 3. A typical PRPD pattern of a corona discharge. Compared to the void discharge, at lower voltage levels the corona discharges occur only during the second half of the cycle, in this case for negative voltages.

lead to different patterns in the PRPD diagram, which are used as the basis for further classification of the defect. The patterns are converted into features, summarizing the pattern itself. A common approach is to bin the data in the phase direction and use the distribution of the number of points, the average, as well as, the maximal discharge strength as features, which are further characterized by typical statistical measures like the skewness and kurtosis (Krivda, 1995b, 1995a). Other approaches interpret the density as an image and make use of methods from image classification.

Partial Discharge Classification has been investigated using almost all approaches developed in machine learning in the past; reviews are given, e.g. in (Danikas, Gao, & Aro, 2003; Sahoo, Salama, & Bartnikas, 2005; Ma, Chan, Saha, & Ekanayake, 2013; Raymond, Illias, Bakar, & Mokhlis, 2015; Barrios, Buldain, Comech, Gilbert, & Orue, 2019; Lu, Chai, Sahoo, & Phung, 2020).

Due to the complex nature of the underlying stochastic process, as well as the large variety of defects and therefore potential patterns, most analysis or classification approaches tend to use data-driven algorithms, based on collecting data from a large number of different discharges either in the lab or in the field and training the algorithm with them.

On the other hand, models to describe partial discharges from a microscopic or physical approach have been explored in the literature as well, see e.g. (Niemeyer, 1995; Cavallini & Montanari, 2006; Callender, Golosnoy, Rapisarda, & Lewin, 2018; Callender & Lewin, 2020).

In many cases the motivation was to develop a deeper understanding of the processes at work. But there were also attempts to use them as a basis for more model-based PD analysis approaches (Heitz, 1999; Altenburger, Heitz, & Timmer, 2002; Cavallini & Montanari, 2006; Patsch & Berton, 2002). In this work we are picking up this way of PD analysis and combining it with modern statistical inference methods.

5. PHYSICS-BASED PARTIAL DISCHARGE MODELING

A simple model to describe the creation of partial discharges is given in this section, following in major parts the one proposed in (Heitz, 1999). This model requires only a rather small set of parameters, which helps with the inference.

In order to reduce the complexity of the approach, we restrict ourselves to a symmetric description, that is, the processes during the positive and negative voltage half-cycles are assumed to be described by the same set of parameters. This leads to a symmetric PRPD pattern, which is not in agreement with many measurements. A more general model allows them to be different for the two half-cycles, doubling their number, but this is needed in order to describe (strongly) asymmetric patterns. As we are mostly exploring the applicability of the ABC approach here, the restriction to symmetric ones seems to be justified.

The discharge within any defect will depend on the local electric field strength. As this field is not directly accessible to us, but can be assumed to be proportional to some related voltage across them, the model converts all electric fields into corresponding voltages. For example, the field inside the defect generated by the applied external line voltage is proportional to $U_{ext}(t) = U_0 \cos(2\pi ft)$. It is convenient to normalize all voltages in the system to be proportional to the amplitude of this line voltage, setting $U_0 = 1$, that is

$$U_{ext}(t) = \cos(2\pi ft). \quad (1)$$

A discharge occurs due to the electrical field exceeding a critical level, corresponding to an “inception voltage” U_{inc} . But the discharge will not occur instantaneously when the total voltage reaches this level; only with a certain probability per time unit, that is a rate $c(t)$. This rate is assumed to originate from two different sources, related to the availability of “seed electrons” assumed to trigger the start of the discharge: A constant source of seed electrons, giving a constant rate c_0 and a time dependent one, with seed electrons being produced during the last discharge, but recombining and therefore disappearing with time. The rate connected to these is given as $c_1 \exp(-t/\tau)$. The total rate is therefore

$$c(t) = \Theta(U_{tot}(t) - U_{inc}) (c_0 + c_1 \exp(-t/\tau)) \quad (2)$$

with the Heaviside or indicator function Θ being one if U_{tot} is larger than U_{inc} and zero otherwise.

The corresponding total voltage inside the defect is given by the external one U_{ext} together with the one produced by the charges created by the individual discharges $U_{int}(t)$, that is

$$U_{tot}(t) = U_{ext}(t) + U_{int}(t). \quad (3)$$

Similarly to the reduction of seed electrons, it is assumed, that if no discharge happens, the internal charges and therefore

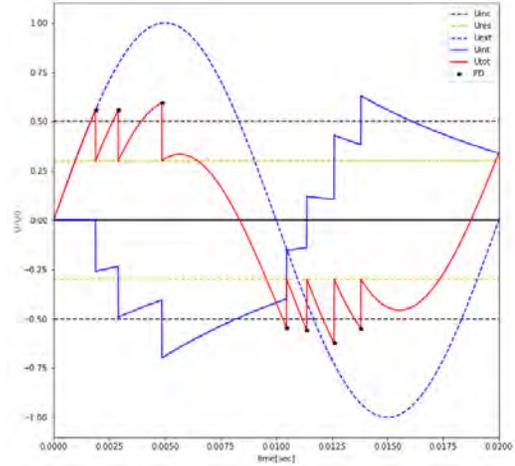


Figure 4. The evolution of the external ($U_{ext}(t)$), total (U_{tot}), and internal (U_{int}) voltage is shown during one cycle. If the total electric field exceeds the inception voltage U_{inc} a discharge can occur with some rate $c(t)$. Such a discharge reduces the voltage to the residual one (U_{res}). Without a discharge the internal voltage is reduced in time.

$U_{int}(t)$ will be reduced with time, described by a decay rate ν , that is

$$\frac{dU_{int}(t)}{dt} = -\nu U_{int}(t) \quad (4)$$

If, on the other hand, a discharge is initiated, charges are flowing and reduce the electrical field and therefore the corresponding voltage U_{tot} . The discharge happens during a short time, until the total electric field, that is the corresponding voltage, reaches the “residual voltage” U_{res} , where it is extinguished

$$U_{tot}^+(t) = U_{ext}(t) + U_{int}^+(t) = U_{res}. \quad (5)$$

Here U^+ denotes the value immediate after the discharge. The total strength of the discharge is assumed to be proportional to the change in voltage

$$\Delta U = U_{tot}^-(t) - U_{tot}^+(t) = U_{int}^-(t) - U_{int}^+(t) \quad (6)$$

where U^- denotes the value immediately before the discharge. The measured discharge strength is given by

$$Q = \gamma \Delta U \quad (7)$$

where the proportionality factor γ is a property of the defect and geometry.

The typical change of U_{tot} and U_{int} with time and the relation with the occurrence of a discharge is shown in Fig. 4.

This description is sufficient to create a simulation code for the generation of discharge event sequences (Q_k, t_k) by following the evolution of the different voltages over time and selecting whether a discharge happens based on the value of $c(t)$. The likelihood function on the other hand is more diffi-

cult to derive, especially for more complex models.

The model has overall seven parameters that need to be determined $\theta = (U_{inc}, U_{res}, c_0, c_1, \tau, \nu, \gamma)$. Despite its rather simple form, the model is able to create quite a number of different PRPD patterns, therefore it can be seen as a model to cover different types of PD defects. The main reason for the variability of patterns is the fact, that there are three time scales in the model: one is related to the AC voltage, corresponding e.g. to 50Hz, one to the reduction of $c(t)$, given by τ , and finally one to the change of $U_{int}(t)$, given by ν . Depending on their respective values or relative sizes the sequence of discharge can vary, e.g. leading to cases with many discharges happening during a half-cycle to some, where a long time between consecutive discharges is present.

6. BASIC PRINCIPLE OF ABC

The basic principle of ABC can be described in a rather simple way: One assumes to have a prior distribution $p(\theta)$ with the capability to generate easily samples from it. Using these model parameters θ_n , it is again assumed that one can generate measurement samples x_n , even though $p(x|\theta)$ is in general not available. The combination (x_n, θ_n) are samples generated from the combined probability density function $p(x, \theta)$.

If one selects from these pairs only those n with $x_n = x$, that is which agree with the observed value x , it is easy to see that the corresponding θ_n are distributed according to the posterior distribution $p(\theta|x)$. In practice such an algorithm is not usable, especially when the measurements x is continuous or high-dimensional, as the agreement $x_n = x$ is hardly ever fulfilled.

Instead one introduces a distance measure between two measurements x and x' : $\rho(x, x')$ and the requirement of $x = x'$ is relaxed to

$$\rho(x, x') \leq \epsilon \tag{8}$$

with ϵ chosen sufficiently small. As the agreement between x and x' is no longer exact, the posterior distribution will be approximate as well.

In many cases the construction of the distance ρ is done with the help of some summary statistics $t(x)$ and the distance is then defined with respect to them, giving

$$\rho(x, x') = \rho(t(x), t(x')) \leq \epsilon. \tag{9}$$

This leads to the simplest approach, the ‘‘ABC rejection algorithm’’, as given in Algorithm 1.

The value of ϵ can often be chosen by running the algorithm with a decreasing series of values until the approximate posterior distribution does not change significantly.

The disadvantage of this algorithm is, that it requires a large

```

for  $n = 1, \dots, N$  do
  do
    Sample  $\theta^* \sim p(\theta)$ 
    Sample  $x^* \sim p(x|\theta^*)$ 
    Calculate  $D = \rho(x, x^*)$ 
  while  $D > \epsilon$ ;
   $\theta_n = \theta^*$ 
end
    
```

Algorithm 1: The ABC rejection algorithm.

number of runs to find a good value of ϵ . In addition, the sampling of θ_n is done with respect to the prior distribution only. This leads to an overall rather inefficient approach. Therefore more refined algorithm, based e.g. on MCMC and SMC have been proposed in the literature (Sisson et al., 2019). In this work we have used the ABC-SMC approach as described in (Toni, Welch, Strelkowa, Ipsen, & Stumpf, 2009). This algorithm uses a decreasing sequence of ϵ_t , either predefined or dynamically adjusted at each step $t = 1, \dots, T$. Details of the algorithm are given in Algorithm 2.

```

Sample  $\theta_n^0 \sim p(\theta), n = 1, \dots, N$ 
Initialize  $w_n^0 = 1/N, n = 1, \dots, N$ 
for  $t = 1, \dots, T$  do
  for  $n = 1, \dots, N$  do
    do
      Sample  $\theta^*$  from  $\theta^{t-1}$  using a multinomial
      distribution with weights  $w^{t-1}$ 
      Perturb  $\theta^*$  to a new  $\theta^{**} \sim Q(\theta, \theta^*)$ 
      Sample  $x^*$  from  $p(x|\theta^{**})$ 
      Calculate  $D = \rho(x, x^*)$ 
    while  $D > \epsilon_t$ ;
     $\theta_n^t = \theta^{**}$ 
  end
  Calculate new weights  $w^t$  using
  
$$w_n^t = \frac{p(\theta_n^t)}{\sum_{n'=1}^N w_{n'}^{t-1} Q(\theta_n^t|\theta_{n'}^{t-1})}$$

  and normalize them
end
    
```

Algorithm 2: The ABC-SMC algorithm.

In the initial step a starting population of θ_n from the prior distribution is chosen and some equal initial weights are initialized. In each subsequent step a value from the current population of parameters is chosen according to its corresponding weight and perturbed using some kernel function $Q(\theta, \theta')$. A value of x is simulated and the new value of θ is kept, if the acceptance criterion $\rho(x, x') \leq \epsilon_t$ is fulfilled. This process is repeated until N new values of parameters θ_n are obtained. The weights are adjusted to account for the prior distribution and the probability of parameters to be chosen based on the former weights and the kernel function. The value of ϵ_t is reduced and the process repeats.

The main advantage of this approach is that ‘‘good’’ values θ_n

are kept and that an automatic reduction of the ϵ_t is done. Refinements with respect to choosing this sequence in an adaptive way and also in order to adjust the kernel with time to improve its width have been proposed in the literature and were implemented and tested as well. Other improvements are possible by keeping only the best results out of a much larger population of parameter values in each step.

Overall these improvement were found to reduce the computational effort and avoid a degeneration of the samples, that is often seen in SMC algorithm (Doucet, De Freitas, Gordo, & James, 2001).

7. APPLICATION OF ABC TO PARTIAL DISCHARGE DATA

In order to apply the ABC algorithm to identify the parameters $\theta = (U_{inc}, U_{res}, c_0, c_1, \tau, \nu, \gamma)$ of the underlying partial discharge model, some additional steps are needed. The basic ABC algorithm requires a measure of the closeness of two discharge patterns x and x' , where each

$$x = \{(t_k, Q_k)\}_{k=1, \dots, K} \quad (10)$$

consists of a sequence of K PD discharge events. Depending on how the measurement is performed, the number of events can be different between x and x' , e.g. if measurements are done during a pre-specified length of time T , or alternatively can be the same, if a fixed number of events is calculated, but this means that the total measurement time T is not the same between them. The total rate of discharges K/T gives important information about the PD type or the parameters of the model. In most practical application of PRPD analysis this is however ignored, and the discharge probability distribution is analyzed instead.

A number of distance measures $\rho(x, x')$ can be chosen in order to compare the two-dimensional point clouds. If one normalizes the PRPD patterns to correspond to a probability density, as commonly done, statistical distances can be used, but they are not suitable for a comparison of the rate densities. Measures could also be based on the features extracted from the PRPD plot, as described in Sec. 4. These have the same disadvantages of being insensitive to the total rate, unless this rate is added to the list of features and also not being comparable in their values. They can also not be extended in a systematic way in most cases.

In this work an approach based on moments of the two-dimensional distribution is used. They are defined both in the phase and discharge-strength direction. As the phase is circular, an expansion in terms of a trigonometric series is done, whereas in the discharge direction, for which we assume that only the absolute value of the discharge strength is measured, power moments are used. The most basic definition of the

moments is

$$M_{i,j} = \frac{1}{T} \sum_k (Q_k)^i [\cos, \sin] (j\phi_k) \quad (11)$$

but for computational efficiency other definitions, e.g normalizing the power series, or using powers of the base trigonometric function instead of the normal Fourier series have been used as well for comparison. Overall no major differences were found with respect to the posterior distribution calculated. Moments up to $i, j = 5$ were used to reduce the computational effort.

The distance measure used is a weighted Euclidean norm

$$\rho(M, M') = \sqrt{\sum_{i,j} w_{i,j} (M_{i,j} - M'_{i,j})^2}. \quad (12)$$

A convenient choice of the weights $w_{i,j}$ was used by bootstrap resampling the measured events and using the (inverse of the) estimated variance of each moment for it. In this way the natural variation of each moment is used and the norm definition is independent of the simulated results.

For the application of ABC to the model uniform priors were chosen for U_{inc} and U_{res} , as these values are required to be positive and are bounded, as too large values make it impossible to have PD events. For all other parameters, which are required to be positive, suitable gamma priors were chosen.

8. RESULTS

The method was applied to both simulated (synthetic) and real data. For simplicity only symmetric PRPD patterns were studied here; the extension to asymmetric ones could be addressed in the same way, but given the fact that the number of parameters almost doubles, this was not done. As a real example, data from a surface discharge measurement was used and was symmetrized before its use.

To confirm in a first step that the approach is suitable in principle and is able to reconstruct the parameters used in the generation of the data, it was applied to synthetic data, generated by the same PD model as used for the analysis. Results of the PRPD pattern for the initial and the final parameters are shown in Fig. 5. The visual convergence of the PRPD patterns is clearly visible. A comparison of the estimated parameters being close to the one used for the simulation is shown in Fig. 6 for one parameter as example. The parameter γ becomes more narrow as the SMC algorithm proceeds.

In the next step, the algorithm was applied to real data. The comparison of the PRPD data was found to be less convincing than in the synthetic case. A more thorough look showed, that there were some difficulties in describing the sharp drop of PD events at the lower end of the pattern.

A deeper investigation revealed that the experimental data

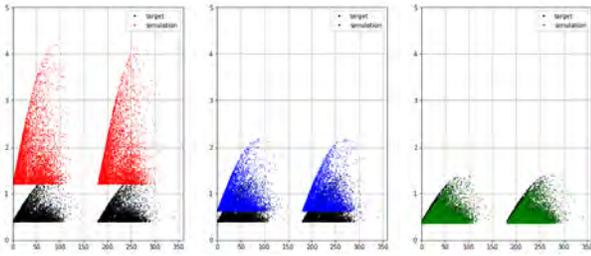


Figure 5. Convergence of the PRPD pattern to the synthetic one during the progression of the ABC-SMC algorithm. Shown are the patterns due to the mean value of the parameter.

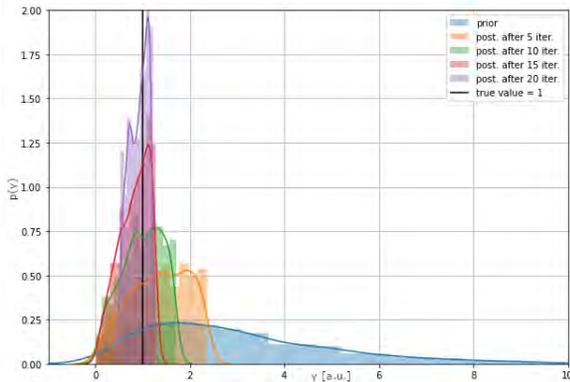


Figure 6. Convergence of the parameter γ to the one used in the generation of the data during the progression of the ABC-SMC algorithm.

had a cutoff introduced, that was not coming from the discharges per se, but from the measurement system, in order to suppress noise. As this threshold is not related to any physical property of the discharge, the PD model will in general not obey it.

The introduction of this threshold on the other hand can be very easily done into the simulation, that is, discharge events are generated and those with a strength below the threshold are removed before the moments are calculated. This shows the very flexible nature of any simulation-based inference, as such measurements effects can be considered in an easy way. It was found, that the ABC algorithm improves the agreement with the measurement in the PRPD plot, if the value is assumed to be known.

We further tested, whether the approach is able to cope with data, where an unknown threshold level exists. This may occur when some automatic pre-processing unit will introduce it without further knowledge of its value. The threshold is introduced as an additional parameter in the model and determined in the same way as the other ones. The ABC algorithm was found to be able to correctly estimate the threshold level. A comparison of the PRPD patterns finally found is shown in

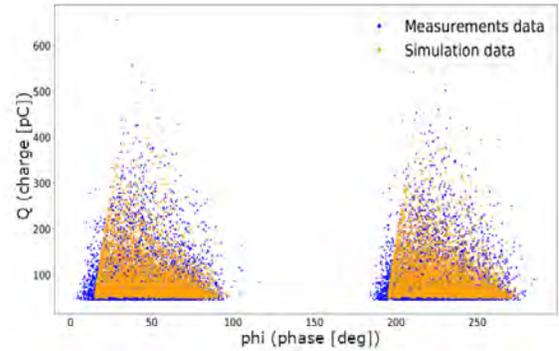


Figure 7. A comparison of a real surface discharge pattern with a cut-off and the result of the ABC algorithm. The threshold was not fixed to the known value in this case, but was determined by the ABC algorithm as well.

Fig. 7. The good agreement at the lower cutoff shows that the introduction of any measurement effects is important to get a good result.

ABC allows even to do further statistical analysis. In Bayesian analysis model comparison can be done in a natural way, using e.g. Bayes factors, or determining the posterior probabilities of e.g. two different models. This can also be implemented in ABC, as described e.g. in (Toni, 2011). This model comparison can be applied to do classification by selecting e.g. priors for the parameters, that are compatible with only a specific type of discharge. Alternatively, it was used here in order to compare different PD models. There have been discussions about the details of the mechanism to generate a specific feature of void discharges, called the “rabbit-ear”, in the literature (Cavallini & Montanari, 2006; Niemeyer, 1995). ABC can be used to compare the two models and decide, which one is more likely to explain the data, even though they have different parameters, as well as even different numbers of them.

The convergence of the probability of each model given the measurement is shown in Fig. 8, whereas a comparison of the predicted PRPD for each model is shown in Fig. 9. Whereas the results are too early to draw already strong conclusions, this shows the potential of ABC even beyond the parameter estimation capabilities.

9. CONCLUSION

This work has shown, that partial discharge analysis can be done with the help of a physical model of the discharge combined with the use of ABC as a simulation-based inference method. A simple physical model was presented and the summary statistics and distance measure needed to define the closeness of the simulated pattern with the measured one was discussed. Starting from the simplest ABC algorithm the use

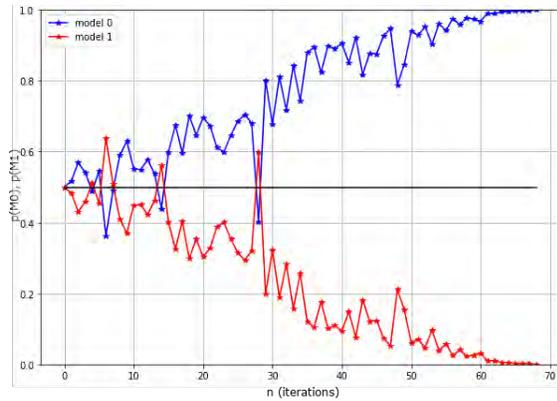


Figure 8. Posterior model probability of two simulations model explaining the origin of the “rabbit ear” phenomena in void discharges. It can be seen that after a number of iterations “model 0” is more likely than “model 1” to explain the PRPD pattern.

of more efficient ones, based on SMC and sequences of accuracy measures is discussed.

An interesting feature of simulation based inference is the possibility to incorporate quite easily phenomena, which are difficult to define in a probabilistic framework. As an example the measurement threshold, which is often applied in practice to reduce the amount of noise, was incorporated easily and it was shown that inferring not only the PD model parameters, but also the threshold itself was possible. Finally the use of ABC for model comparison was shown, which has a direct link to PD classification. Alternatively, it can be used as a method to distinguish between different PD models discussed in the literature.

Simulation based inference of partial discharge can be seen as a follow-up of research activities done in the 90s, in contrast to data-driven methods explored more recently. One limitation at that time was clearly the available computing power, which required quite specialized techniques. With the increase of computing power, but also the development of modern inference techniques this approach to PD analysis seems to be within reach today.

Simulation-based inference should be seen as an interesting approach to diagnostics and prognostics beyond the application to PD analysis. The detailed knowledge of the technical devices monitored are an import input, that should be taken into account. In addition, the uncertainty quantification of the parameters extracted, but also the large number of “hidden variables” that are not relevant for the degradation of the system, can both be dealt with. Therefore the application of this method to other systems is clearly of interest, and should be explored in the future.

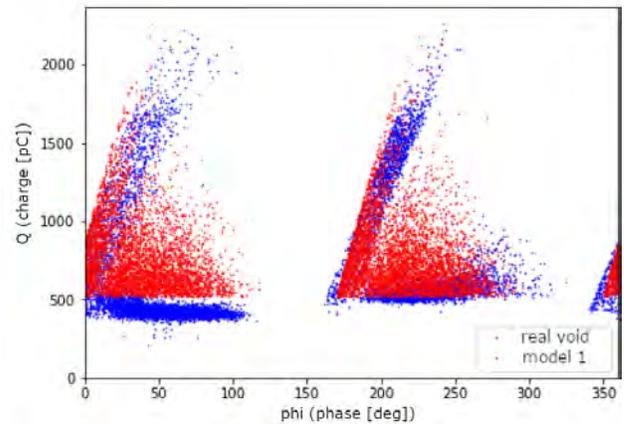
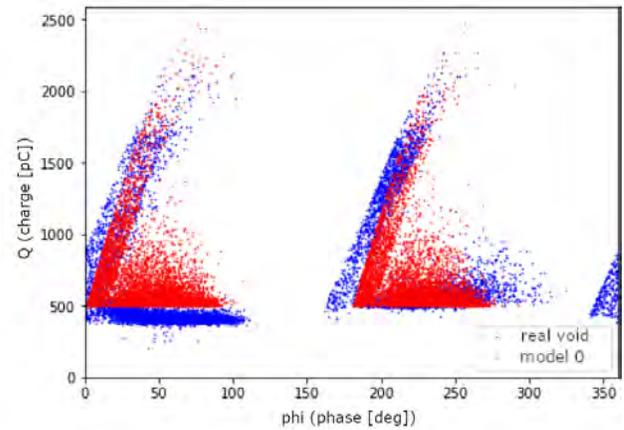


Figure 9. A comparison of the PRPD pattern as generated by the two models in the final iteration and compared to the real data. The upper plots shows the result for model0, the lower one for model 1.

ACKNOWLEDGMENT

This work has profited from a large number of discussions with different people within the research center and within ABB. Especially the insight of Yannick Maret at ABB on signal processing and data analysis, as well as Thomas Christen and Jörg Lehmann, both now at Hitachi Energy, into the foundations of electrical insulation, partial discharges, and the theory of stochastic processes have helped at various stages of the work.

REFERENCES

Altenburger, R., Heitz, C., & Timmer, J. (2002). Analysis of phase-resolved partial discharge patterns of voids based on a stochastic process approach. *Journal of Physics D: Applied Physics*, 35(11), 1149.

Barrios, S., Buldain, D., Comech, M. P., Gilbert, I., & Orue, I. (2019). Partial discharge classification using deep learning methods—survey of recent progress. *Energies*, 12(13), 2485. doi: 10.3390/en12132485

- Beaumont, M. A. (2019). Approximate bayesian computation. *Annual Review of Statistics and Its Application*, 6(1), 379-403.
- Callender, G., Golosnoy, I. O., Rapisarda, P., & Lewin, P. L. (2018). Critical analysis of partial discharge dynamics in air filled spherical voids. *Journal of Physics D: Applied Physics*, 51(12), 125601.
- Callender, G., & Lewin, P. L. (2020). Modeling partial discharge phenomena. *IEEE Electrical Insulation Magazine*, 36(2), 29–36.
- Cavallini, A., & Montanari, G. C. (2006). Effect of supply voltage frequency on testing of insulation system. *IEEE transactions on Dielectrics and Electrical Insulation*, 13(1), 111–121.
- Cranmer, K., Brehmer, J., & Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48), 30055-30062.
- Csilléry, K., François, O., & Blum, M. G. (2012). abc: an R package for approximate bayesian computation (abc). *Methods in ecology and evolution*, 3(3), 475–479.
- Danikas, M. G., Gao, N., & Aro, M. (2003). Partial discharge recognition using neural networks: A review. *Electrical Engineering*, 85(2), 87–93.
- Doucet, A., De Freitas, N., Gordo, & James, N. (2001). *Sequential monte carlo methods in practice* (Vol. 1) (No. 2). Springer.
- Dutta, R., Schoengens, M., Pacchiardi, L., Ummadisingu, A., Widmer, N., Künzli, P., ... Mira, A. (2021). ABCpy: A high-performance computing perspective to approximate bayesian computation. *Journal of Statistical Software*, 100(7), 1–38.
- Heitz, C. (1999). A generalized model for partial discharge processes based on a stochastic process approach. *Journal of Physics D: Applied Physics*, 32(9), 1012.
- International Electrotechnical Commission (IEC). (2000). *High-voltage test techniques - partial discharge measurements* (Tech. Rep. Nos. 60270:2000+AMD1:2015 CSV). Geneva, Switzerland: IEC.
- Krivda, A. (1995a). Automated recognition of partial discharges. *IEEE Transactions on Dielectrics and Electrical Insulation*, 2(5), 796–821.
- Krivda, A. (1995b). *Recognition of discharges: Discrimination and classification* (PhD thesis). Technical University Delft.
- Lu, S., Chai, H., Sahoo, A., & Phung, B. T. (2020). Condition monitoring based on partial discharge diagnostics using machine learning methods: A comprehensive state-of-the-art review. *IEEE Transactions on Dielectrics and Electrical Insulation*, 27(6), 1861–1888. doi: 10.1109/TDEI.2020.009070
- Lv, Z., Rowland, S. M., Chen, S., Zheng, H., & Iddrissu, I. (2017). Evolution of partial discharges during early tree propagation in epoxy resin. *IEEE Transactions on Dielectrics and Electrical Insulation*, 24(5), 2995-3003.
- Ma, H., Chan, J. C., Saha, T. K., & Ekanayake, C. (2013). Pattern recognition techniques and their applications for automatic classification of artificial partial discharge sources. *IEEE Transactions on Dielectrics and Electrical Insulation*, 20(2), 468–478.
- Montanari, G. (1995). Aging and life models for insulation systems based on pd detection. *IEEE Transactions on Dielectrics and Electrical Insulation*, 2(4), 667-675.
- Niemeyer, L. (1995). A generalized approach to partial discharge modeling. *IEEE transactions on Dielectrics and Electrical insulation*, 2(4), 510–528.
- Patsch, R., & Berton, F. (2002). Pulse Sequence Analysis - a diagnostic tool based on the physics behind partial discharges. *Journal of Physics D: Applied Physics*, 35(1), 25–32.
- Raymond, W. J. K., Illias, H. A., Bakar, A. H. A., & Mokhlis, H. (2015). Partial discharge classifications: Review of recent progress. *Measurement*, 68, 164–181. doi: 10.1016/j.measurement.2015.02.032
- Sahoo, N., Salama, M., & Bartnikas, R. (2005). Trends in partial discharge pattern classification: A survey. *IEEE Transactions on Dielectrics and Electrical Insulation*, 12(2), 248–264. doi: 10.1109/TDEI.2005.1430395
- Sisson, S. A., Fan, Y., & Beaumont, M. A. (Eds.). (2019). *Handbook of approximate bayesian computation*. CRC Press.
- Tavaré, S., Balding, D. J., Griffiths, R. C., & Donnelly, P. (1997, 02). Inferring Coalescence Times From DNA Sequence Data. *Genetics*, 145(2), 505-518.
- Toni, T. (2011). Abc smc for parameter estimation and model selection with applications in systems biology. *Nature Precedings*, 1–1.
- Toni, T., Welch, D., Strelkowa, N., Ipsen, A., & Stumpf, M. P. H. (2009). Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of Royal Society Interface*, 6, 187–202.
- Wang, L., Cavallini, A., Montanari, G. C., & Testa, L. (2010). Patterns of partial discharge activity in xlpe: From inception to breakdown. In *2010 10th ieee international conference on solid dielectrics* (p. 1-4).

BIOGRAPHIES



Kai Hencken is a senior principal scientist for “Physical and Statistical Modeling” at the ABB Corporate Research Center, Baden-Dättwil, Switzerland. He received the Diploma in Physics (1990) and the Ph.D in Theoretical Physics (1994) from the U Basel. He was a post-doc at the Institute for Nuclear Theory, U Washington, Seattle, USA from 1995 to 1997 and at the University of Basel from

1997 until 2005, where he received his Habilitation in 2000 and is a lecturer since. In 2005 he joined the theoretical physics group at the ABB Corporate Research Center. His research interests are the combination of physical modeling with advanced statistical methods to solve problems related to industrial devices. A strong focus is on developing diagnostics and prognostics approaches.



Daniele Ceccarelli is a Data Scientist at Ospedale San Raffaele, Milano, Italy. He received the Bachelor in Mathematical Engineering in 2018 and the Master in Mathematical Engineering in 2021 from the Politecnico di Milano, Milano, Italy. In 2021 he worked as Intern at the ABB Corporate Research Center, Baden-Dättwil, Switzerland.

His research interests are Bayesian statistics, Data Science and Physics-Informed Machine Learning.



Elsi-Mari Borrelli is a Lead Quantum Biology Researcher at Algorithmiq since 2022. She received the Master in Physics in 2009 and a Ph.D. in Theoretical Physics in 2013 from the University of Turku, Turku, Finland. She was a post-doc at Aalto University, Aalto, Finland from 2013 to 2015 and at the University of Turku, Turku, Finland from 2015-2017. From 2017 to 2021 she worked at the ABB

Corporate Research Center, Baden-Dättwil, Switzerland. Her research interests range from advanced analytics methods for industrial devices to application of quantum algorithms for complex optimization problems.



Andrej Krivda is a principal scientist at the ABB Corporate Research Center, Baden-Dättwil, Switzerland. He received the M.Sc. degree from the Kosice University of Technology, Kosice, Slovakia, and the Ph.D. degree from the Delft University of Technology, Delft, The Netherlands. He was with the Royal Institute of Technology, Stockholm, Sweden, for one year, where he worked on automated recognition of partial discharge patterns in electrical power generators. He was four years with the Queensland University of Technology, Brisbane, Australia, where he studied outdoor insulation for overhead power lines. Since 2001, he has been with ABB Switzerland Ltd., Corporate Research, Baden-Dättwil, Switzerland. His current research interests include partial discharges, dielectric spectroscopy, outdoor insulation and intelligent systems for recognition of digital partial discharge patterns. Dr. Krivda was awarded the Power Engineering Journal and the Journal of Electrical and Electronics Engineering, Australia, premiums for his work on overhead insulated mains in 2000.

His current research interests include partial discharges, dielectric spectroscopy, outdoor insulation and intelligent systems for recognition of digital partial discharge patterns. Dr. Krivda was awarded the Power Engineering Journal and the Journal of Electrical and Electronics Engineering, Australia, premiums for his work on overhead insulated mains in 2000.

Unsupervised Prognostics based on Deep Virtual Health Index Prediction

Martin Hervé de Beaulieu¹, Mayank Shekhar Jha², Hugues Garnier³ and Farid Cerbah⁴

^{1,2,3} *Université de Lorraine, CRAN, CNRS UMR 7039, 54506 Vandoeuvre-les-Nancy, France*

martin.herve-de-beaulieu@univ-lorraine.fr

mayank-shekhar.jha@univ-lorraine.fr

hugues.garnier@univ-lorraine.fr

⁴ *Dassault Aviation 92552 Saint-Cloud, France*

Farid.Cerbah@dassault-aviation.com

ABSTRACT

Prediction of the Remaining Useful Life (RUL) for industrial systems has been facilitated by the acquisition of large amounts of real-time data and the use of deep learning methods. However, the vast majority of these methods rely on the availability of extensive RUL-labeled data, which is not the case for most of real industrial applications. The goal of this paper is to show how unsupervised learning can provide alternative ways to address this issue. The proposed method is essentially made of two steps. First, a Virtual Health Index (VHI) is extracted in an unsupervised manner from the raw sensor data using a Deep Convolutional Neural Network (CNN) autoencoder. Secondly, an Long-Short Term Memory (LSTM) Encoder-Decoder predicts the future values of the VHI, until an End-of-Life (EOL) pattern is recognized (using a sliding window DTW algorithm). The suggested method is tested on the C-MAPSS dataset and offers promising results with a great potential to be applicable on real-life use cases.

1. INTRODUCTION

The increase in data collection and storage capabilities has led to the use of deep learning methods in predictive maintenance strategies. Two main indicators are used in the Prognostic and Health Management (PHM) community (Lee et al., 2014). The first is the Health Index (HI). Health Index is an indicator which represents the State of Health (SOH). It is built from the measured data collected by the sensors placed on the system (Lei et al., 2018). It is a critical indicator, as it should reveal the degradation process hidden within the different signals. A common approach for constructing such an HI from multiple signals is to fuse them into a single indica-

tor, using for example a deep neural network based compression technique (C. Hu, Youn, Wang, & Yoon, 2012). Such an HI is called “Virtual Health Index” (VHI). It worth noting that a VHI is an implicit representation of the degradation but it cannot be interpreted from a physical point of view.

Autoencoder models are very efficient in extracting VHI from raw sensors data, being well adapted for the treatment of multivariate non-stationary data. It has been shown that autoextracted features are preferable over the handcrafted features in the case of bearing vibrations (Y. Hu, Palmé, & Fink, 2016), exhibiting monotonicity and clear trendability. Such an AI-based extraction of VHI can then be employed to perform RUL prediction (Gensler, Henze, Sick, & Raabe, 2016), in a two-stage framework similar to the method we present in this paper. Moreover, CNN is a particular deep learning model which has shown great ability on feature extraction, especially in the domain of image classification, speech recognition and time series prediction (LeCun, Bengio, et al., 1995). CNN has also been successfully applied to health monitoring and prognostics (Babu, Zhao, & Li, 2016) (Li, Ding, & Sun, 2018), in the context of direct mapping of raw sensor data to health state indicators. Therefore, there is good reason to think that the combination of CNN and autoencoders would offer very good VHI extraction capabilities from raw sensor data.

The second important indicator in PHM is the Remaining Useful Life (RUL) which is defined as the remaining operating time before the End Of Life (EOL) of the system is reached (Si, Wang, Hu, & Zhou, 2011). A large number of contributions have been published in the recent years in order to propose methods for RUL prediction, based on deep learning approaches, using various types of neural networks. The vast majority of these papers relies on the direct mapping between the sensor values and the RUL prediction. These ap-

Martin Hervé de Beaulieu et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

proaches are supervised and thus require large RUL-labeled dataset for training. However, in practice, these labels are, most of the time, not available. In fact, to obtain the genuine value of the RUL, it is necessary to take measurements of the degradation level until the EOL of the machinery. This is time consuming and extremely costly, as it requires numerous (possibly complete) failure tests. Consequently, such methods, although theoretically sound, are hardly applicable in an industrial environment. A few projects have already been carried out with a perspective of avoiding of labeled data. A RUL estimation method based on a Health Index obtained by using the reconstruction error has been proposed (Malhotra et al., 2016). In this paper, an LSTM-based encoder-decoder is trained to reconstruct the time series corresponding to the healthy state of the system. With increasing degradation, the reconstruction error also increases, resulting in a Health Index. Other suggestions also involve Unsupervised Kernel Regression (UKR) (Khelif, Malinowski, Chebel-Morello, & Zerhouni, 2014) or Adversarial Regressive Domain Adaptation (ARDA) (Jiang, Xia, Wang, Fang, & Xi, 2022) for RUL prediction.

The current paper addresses the problem of RUL prediction in the absence of RUL-labeled data. To this end, an unsupervised method of RUL prediction in two steps is proposed. First, a VHI is extracted using a pruned CNN autoencoder in a non-supervised manner. Secondly, long-range prediction of this VHI is performed by an LSTM encoder-decoder. Such a structure leverages the recurrent nature of the data, and has already proven its efficiency on similar use cases (Yu, Kim, & Mechefske, 2019). Prediction continues until an end-of-life pattern is recognized. For this recognition task, Dynamic Time Warping (DTW) similarity measure is employed. As a result, the RUL can be estimated without using any labeled data during the training phase. This paper extends and improves upon our previous work (Herve de Beaulieu, Jha, Garnier, & Cerbah, 2022), incorporating two major new contributions, namely: the VHI extraction using a CNN autoencoder and the EOL pattern recognition with DTW.

The rest of the paper is organized as follows. The problem statement and background are presented in Section 2. The architecture of the proposed method is introduced in Section 3. Application results to the C-MAPSS dataset, including technical choices and data processing are presented in Section 4. Conclusion and future work are discussed in Section 5.

2. PROBLEM STATEMENT AND BACKGROUND

In this section, the fundamental background necessary for the understanding of the proposed approach will be briefly introduced.

2.1. Problem statement

Let us suppose that we have N identical category equipment, e.g. N engines with index $1 \leq i \leq N$ for which we collect data from K sensors with index $1 \leq k \leq K$. Each data record is made until the End of Life (EOL), denoted as T_i , of the equipment is reached, with index $1 \leq t \leq T_i$. The set of data is thus a collection of objects $\{X^{(i)} | i \in [1, N]\}$ with each data sample $X^{(i)} \in \mathbb{R}^{T_i \times K}$. Therefore, the k -th column $X_k^{(i)}$ corresponds to the vector of values of sensor k for all time steps $1 \leq t \leq T_i$ and the t -th row $X^{t(i)}$ corresponds to the vector of values of all sensor $1 \leq k \leq K$ for the given time step t . Finally, the scalar $X_k^{t(i)}$ is the single value recorded by sensor k at time step t on equipment i . The problem can then be formulated as follows: from an initial subset (or window) $X^{t_1(i)}$ to $X^{t_2(i)}$ with $1 \leq t_1 \leq t_2 \leq T_i$, the corresponding set of VHI values $VHI^{t_1(i)}$ to $VHI^{t_2(i)}$ must be extracted. Based on this VHI set, the RUL value for the last time step of the window t_2 , denoted as $RUL^{t_2(i)}$ must be predicted.

2.2. CNN AutoEncoder

An encoder-decoder is a neural network structure which is composed of two elements. First, an encoding function f_{θ_e} compresses the input data to a subspace called latent space. Second, the compressed representation is expanded through a decoding function g_{θ_d} . The learning process can then be formalized as follows:

$$y = g_{\theta_d}(f_{\theta_e}(x)). \quad (1)$$

with y the output of the encoder-decoder and x the input data. A special case of encoder-decoder is the autoencoder, where the output y is learned to be the reconstruction of the input x , which we note $y = x'$. The loss (also called “reconstruction error”) is therefore obtained via a function of x and x' :

$$J_{AE}(\theta_e, \theta_d) = \sum L(x, x') = \sum L(x, g_{\theta_d}(f_{\theta_e}(x))) \quad (2)$$

where L is a loss function such as the mean squared error (Bengio, Courville, & Vincent, 2013). Note that an autoencoder is trained in an unsupervised manner since the cost calculation does not require any labeled data.

CNN autoencoder is a particular case of autoencoder where the encoding and decoding functions are achieved by CNN structures. It consists of performing a convolution product between the input I and a kernel F (also called filter). For time series, a one-dimensional convolution is applied along the time. The expression of the process is as follows:

$$S(t) = (I * F)(t) = \sum_{t=1}^T I(t)F(t - T) \quad (3)$$

with S the output (also called feature map), I the input, F the kernel, T the overall length of the input and $*$ the convolution

product (Goodfellow, Bengio, & Courville, 2016).

Since the time series are K -dimensional, the K different variables (e.g. K different sensors) are treated as multiple channels. An independent kernel is thus assigned to each channel, resulting in K feature maps. By varying the number of kernels, the dimensionality of the data can then be extended or reduced. In particular, in the case of CNN autoencoders, care should be taken to reduce the number of dimensions to obtain a latent space of reduced size.

2.3. LSTM Encoder-Decoder

Recurrent neural network (RNN) is a neural network structure adapted to sequential learning. It uses prior knowledge along with current input to make the prediction of the desired output. Therefore, RNN models are widely used for sequential data learning such as time series. A recurrent model is made of multiple standard cells whose states are affected by both past states and current input. The most used version of this kind of cell is called “Long-Short Term Memory” (LSTM) (Hochreiter & Schmidhuber, 1997). LSTM have been widely used for RUL prediction as well (Wu, Yuan, Dong, Lin, & Liu, 2018), (Zhang, Zhang, Shao, Niu, & Yang, 2020).

Similarly to CNN autoencoder, LSTM encoder decoder is nothing else than a special case of encoder-decoder where the encoding and decoding functions are performed by RNN models (Cho et al., 2014). Therefore, the RNN-encoder transforms the input time series into a fixed-dimensional vector representation (usually referred to as “context vector”) while the RNN-decoder maps this context vector to the target time series.

For a univariate source time series $X = \{x_1, x_2, \dots, x_T\}$, h_t is the hidden state of the RNN-encoder at time t . The RNN-encoder captures relevant information as the source time series is scanned and when its last time step T is reached, the hidden state h_T is the vector representation of the entire source time series X . The RNN-decoder, in a mirroring operation, takes the final encoding hidden state h_T as initial decoding hidden state. It then constructs the desired output time series. The desired output can be a reconstruction of the input ($X' = \{x'_1, x'_2, \dots, x'_T\}$), in this case it is called RNN-autoencoder and it is an unsupervised process. The output can also be a prediction of the future values of the source time series (Du, Li, Yang, & Horng, 2020). Then, it is a supervised learning task requiring a training set containing the labels of future time series values.

Regardless of the type of output chosen, the model is trained to minimize an error (which is called “reconstruction error” in the case of RNN-autoencoder) between the target and the

result of the model. It can be written as:

$$E = \sum_{i=1}^N \sum_{t=1}^T (y_t - y'_t) \quad (4)$$

for N time series of length T , with y_t the target value and y'_t the value obtained from the model.

Different recurrent structures and options can be used for the encoding and decoding process, such as stacked-RNN and/or bidirectional RNN (Yu et al., 2019).

2.4. Sliding DTW pattern recognition

Dynamic Time Warping (DTW) is a similarity measuring technique for time series traditionally used in speech recognition (Sakoe & Chiba, 1978). Let us consider two time series $X = (x_1, x_2, \dots, x_N)$, and $Y = (y_1, y_2, \dots, y_M)$ of lengths N and M . A local cost matrix C representing all pairwise distances between X and Y is built. To do so, a local cost measure c is computed between each pair of elements of the sequences X and Y . Thus, the local cost matrix can be denoted as $C \in \mathbb{R}^{N \times M}$ and is defined by $C(n, m) = dist(x_n, y_m)$ with $n \in [1 : N]$, $m \in [1 : M]$ and $dist$ being a local distance measure (e.g. $dist(x_n, y_m) = \|x_n - y_m\|$).

The goal is then to find the alignment path which runs through the low-cost areas of the local cost matrix. A warping path is formally defined as a sequence of points $p = (p_1, p_2, \dots, p_L)$ with $p_l = (p_n, p_m) \in [1 : N] \times [1 : M]$ for $l \in [1 : L]$. Any warping path must respect three conditions:

1. *Boundary condition:* $p_1 = (1, 1)$ and $p_L = (N, M)$. This basically means that the starting and ending points of the warping path are effectively the first and the last points of the two sequences X and Y .
2. *Monotonicity condition:* $n_1 \leq n_2 \leq \dots \leq n_L$ and $m_1 \leq m_2 \leq \dots \leq m_L$. This condition ensures that the points are correctly time-ordered.
3. *Step size condition:* $p_{l+1} - p_l \in \{(1, 0), (0, 1), (1, 1)\}$ for $l \in [1 : L - 1]$. This criterion prevents the warping from doing big shifts in time while aligning the two sequences.

The total cost of a warping path p is expressed as:

$$C_p(X, Y) = \sum_{l=1}^L dist(x_{n_l}, y_{m_l}) \quad (5)$$

The DTW distance between two time series X and Y is then the total cost of the optimal warping path denoted as p^* , which is the warping path having the minimal total cost among all the possibilities. The major benefit of DTW is that it minimizes the effects of shifting and distortion in time by allowing a flexible transformation of time series with the intention of detecting similar shapes. A simplified comparison between euclidean and DTW distances is illustrated in Figure 1. The Python library used in this work is from (Giorgino, 2009).

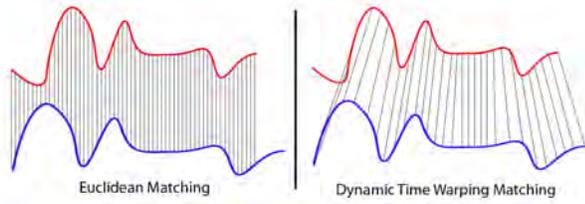


Figure 1. Simplified comparison between the Euclidean distance and the DTW distance.

The DTW is used in the proposed approach in order to recognise a pattern Y in a time series X . Therefore, DTW is applied by using a sliding time window mechanism. A window denoted $W(X)$ (i.e. a sub-sequence of X) is sliding over the full sequence X , with a stride s . At each step, the DTW distance is computed between the sliding window $W(X)$ and the pattern (i.e. template time series) Y .

3. PROPOSED METHOD

The main advantage of the method proposed in this paper is to provide a prediction of RUL in an unsupervised way, i.e. without needing RUL-labeled data. To do this, we proceed in two steps.

Step 1: a CNN autoencoder model is used to extract from the raw sensor data a univariate Virtual Health Index (VHI) which is a compression of the multi-variate input data. The excellent feature extraction abilities of such a structure enable to highlight a so called “End-of-Life-pattern” which characterizes the moment when the EOL of the studied equipment occurs.

More specifically, for the suggested method, the CNN encoder model is made of 9 convolution layers. In order to extract deep features, the first layers increase the depth of the data by expanding the number of channels from $K = 7$ to $K = 56$ and then compress it to obtain a univariate VHI in the latent space ($K = 1$). The CNN decoder is built as a mirror of the encoder. The hyper parameters of the convolutions layers are chosen so that the initial length of the time series remains unchanged over the convolution operations. Specifically : stride $s = 1$; kernel length $k_l = 23$; symmetrical padding $p = 11$. Between each convolution, a ReLU activation function is apply to ensure non-linearity. Figure 2 shows the overall CNN autoencoder structure.

Step 2: an LSTM encoder-decoder is used in order to predict the future values of the VHI. This structure is made of deep-stacked LSTM (3 layers of 120 hidden units) whose role is to extract the deep temporal features of the VHI time series and to output a prediction window of length P . This has already been presented in (Herve de Beaulieu et al., 2022). The final objective is to estimate the RUL. To do so, for each prediction window, the presence of the EOL pattern is checked. As long as the pattern announcing the end of life is not detected,

the prediction continues, using previously predicted values as input for subsequent predictions. The detection of the EOL-pattern is achieved by measuring the DTW distance between the prediction windows and the EOL-pattern that have been extracted by the CNN autoencoder. The prediction continues step by step until the EOL-pattern is recognized with a sufficiently high similarity (meant as a threshold). When that time is reached, this means that the equipment has reached its EOL. The RUL can thus be inferred recursively by counting the number of prediction cycles that were necessary. The proposed RUL inference process is summarized by Figure 3. Min-similarity threshold, stride and window lengths are set initially in an empirical manner and the optimized by using a validation set.

Of course, the closer the input data is to the end of life, the lower the length to be predicted and therefore the more accurate the deduced RUL. This results in a higher variability in the early predictions (temporally distant from the EOL) than in the latter.

Input: VHI window of length T denoted as X , EOL pattern denoted as Z .

Output: RUL value (scalar).

$i = 0$;

$Y \leftarrow$ VHI predicted window of length P from X ;

while $DTW(Y, Z) > threshold$ **do**

$i \leftarrow i + 1$;

$X \leftarrow X[S:] + Y[0:S]$; /* stride S */

$Y \leftarrow$ new VHI predicted window from new input X ;

end

$RUL \leftarrow S \times i + P$;

Figure 3: RUL prediction process for one input VHI window

4. EXPERIMENTAL RESULTS

4.1. C-MAPSS dataset

The C-MAPSS dataset is used as experimental data to test the proposed method. This dataset is composed of turbofan engines degradation trajectories which are obtained from a simulator developed by NASA (Saxena, Goebel, Simon, & Eklund, 2008). C-MAPSS dataset is divided into four different subsets (named FD001 to FD004), each made of a training set containing several complete degradation data (i.e. multi-variate data collected from sensors) and a test set containing truncated degradation data (from the same distribution as training set), the objective being to predict the RUL based on this incomplete test data. Depending on the subset of C-MAPSS, fault modes can vary from one in FD001 and FD002 to two in FD003 and FD004. Similarly, the data is obtained from simulations carried out under a variable number of operating conditions. These conditions are based on different combinations of altitude (0 to 42000 feet), throttle resolver angle (20 to 100) and Mach (0 to 0.84). More details about

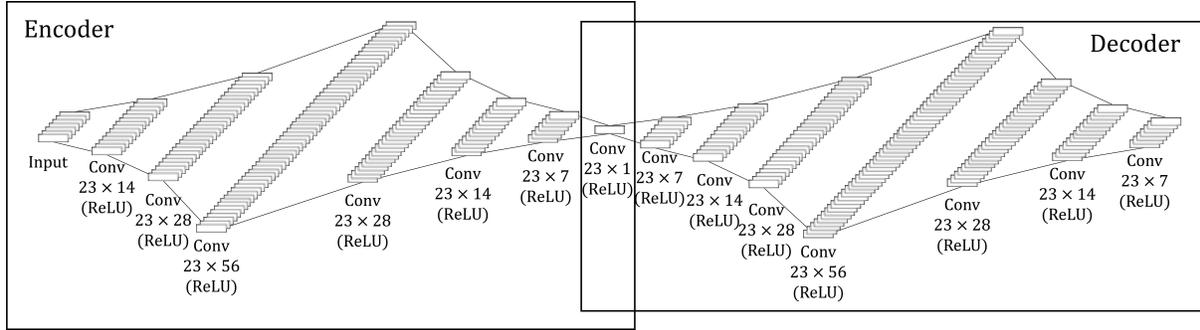


Figure 2. Proposed deep CNN autoencoder structure for VHI extraction.

Table 1. Features of the C-MAPSS dataset.

	C-MAPSS subsets			
	FD001	FD002	FD003	FD004
Engine for training	100	260	100	249
Engine for testing	100	259	100	248
Operating conditions	1	6	1	6
Fault modes	1	1	2	2

the different subsets are given in Table 1.

4.2. Data preparation

21 different sensor variables are available in the C-MAPSS dataset. To reduce the computational burden, only the seven most indicative sensors are kept, based on the observation of the input time series (Wang, Yu, Siegel, & Lee, 2008). Therefore, the input set is composed of sensors number 2, 3, 4, 7, 11, 12, 15. The detailed list of sensors, along with the units and description, is available in Appendix A. All sensor time series are normalized following a standard scaling defined as:

$$Norm(x^s) = \frac{x^s - \mu^s}{\sigma^s} \quad (6)$$

where s is the selected sensor, μ^s and σ^s are the mean and the standard deviation.

Before providing this data to the CNN autoencoder, a zero pre-padding operation is applied, in order to force all time series to have the same length (Dwarampudi & Reddy, 2019). Once the VHI has been obtained, the univariate VHI is un-padded to go back to its original length. Figure 4 gives an example of padded and normalized multi-variate sensor time series given as input to the CNN autoencoder.

4.3. Performance indicator of the RUL prediction

The evaluation of the performance of the RUL prediction made by the proposed method is handled using a Root Mean Square Error (RMSE). Indeed, it is the most used performance indicator in RUL prediction literature, especially on C-MAPSS. In the future, it may be considered to complete this metric with other indicators such as the score function (Saxena et

al., 2008) or the Mean Absolute Percentage Error (Malhotra et al., 2016). RMSE is computed as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n d_i^2}, \quad (7)$$

for n predictions, where d_i is the difference between the predicted and the actual RUL:

$$d_i = R\hat{U}L_i - RUL_i. \quad (8)$$

4.4. Results

4.4.1. Step 1 - VHI extraction using CNN autoencoder

At the end of the training phase, an EOL pattern is recognised by the deep CNN autoencoder. It is a local minimum whose shape and position are always identical for all the turbines of the training set. Let us keep in mind that, as mentioned in Section 1, VHI does not have any physical interpretation. It should be considered as an indication of the EOL (based on the location of the pattern), not as a physical measure of the equipment SOH. Therefore, the fact that the VHI increases at the end of life does not indicate an improvement in SOH. Such an EOL pattern can be seen in Figure 4 and in Figure 5. On the latter, the true RUL has also been plotted in order to show that the pattern is effectively corresponding to the EOL. Therefore, this pattern will serve as template to be detected on the test data. Note that after having obtained the VHI, the zero-pre-padding is removed to keep the original length. As explained in section 4.1, the test set is composed of time series which are not reaching the end of life. Thus, this test data does not yet reveal the pattern.

4.4.2. Step 2 - VHI long-range prediction and RUL inference

Therefore, from the unfinished test VHI, future predictions are obtained by the LSTM encoder-decoder. As described in Section 3, future VHI values are predicted in a rolling window process, re-using the previous predictions as new inputs for the next ones. Figure 6 shows the result of such a predic-

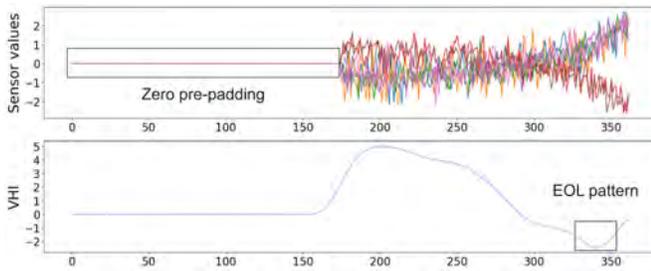


Figure 4. Top: an example multi-variate input data for one turbine from the training set. Bottom: the corresponding VHI extracted by the CNN autoencoder.

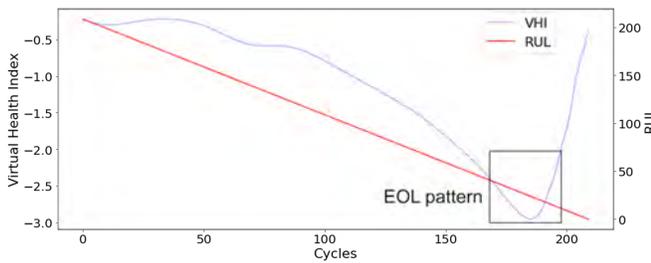


Figure 5. An example of extracted VHI (dotted line) along with RUL labels of one turbine from the training set.

tion process for one turbine, from time step $t = 45$. Here, the whole process of prediction only relies on the 50 values preceding the time step $t = 45$. The rest of the prediction process is self-feeding, by reusing past predictions. On the same figure is displayed the reference pattern, at the instant where it has been recognized with a satisfying DTW distance score value, using the sliding DTW algorithm introduced in Section 2.4. This value is set empirically, in a hyperparameter optimization loop conducted on a validation set. The VHI prediction process is applied for each turbine available, at each time step, thus resulting for each turbine in a sequence of RUL values. Such a trajectory is shown in Figure 7, for the same turbine as in Figure 6. The RMSE of the RUL trajectory is calculated for each turbine of the test set, leading to an average RMSE of 40.1 and a standard deviation of 21.7.

5. CONCLUSION

An unsupervised RUL prediction method has been proposed in this paper that avoids the dependence on RUL-labeled data, therefore offering great interest for real-world applications. A VHI has been extracted from sensor data in an unsupervised manner, using a deep CNN autoencoder, highlighting a clear end-of-life pattern. Then, using an LSTM encoder decoder, future values of unseen VHI have been predicted, until the EOL pattern is recognized using a DTW distance measure. This pattern detection algorithm allows to deduce precisely the RUL value from the VHI without needing the RUL labels. The proposed method has been tested with success on the C-MAPSS dataset.

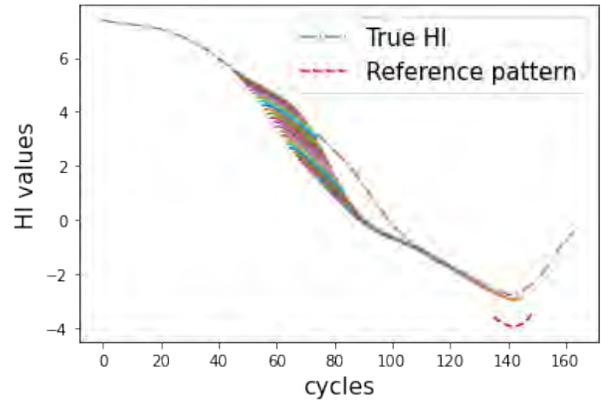


Figure 6. Overview of the set of predicted sliding windows from time step $t = 45$, until the reference pattern (in red) is recognized with a high enough DTW measure of similarity.

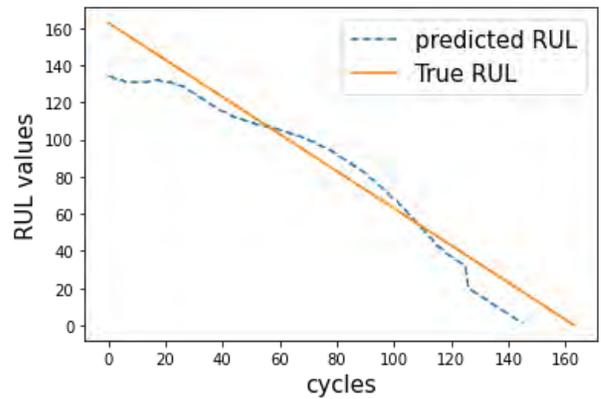


Figure 7. Complete RUL trajectory for all the time steps for one turbine.

As a perspective, this unsupervised RUL prediction problem will be comprehensively extended to handle various conditions and/or various fault modes. In particular, the other datasets included in C-MAPSS will be used, as they offer up to 6 different operating conditions based on 3 variable settings mixed randomly during each flight and up to 2 fault modes.

REFERENCES

Babu, G. S., Zhao, P., & Li, X.-L. (2016). Deep convolutional neural network based regression approach for estimation of remaining useful life. In *International conference on database systems for advanced applications* (pp. 214–228). Dallas, Texas, USA.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798–1828.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D.,

- Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Du, S., Li, T., Yang, Y., & Horng, S.-J. (2020). Multivariate time series forecasting via attention-based encoder-decoder framework. *Neurocomputing*, 388, 269–279.
- Dwarampudi, M., & Reddy, N. (2019). Effects of padding on LSTMs and CNNs. *arXiv preprint arXiv:1903.07288*.
- Gensler, A., Henze, J., Sick, B., & Raabe, N. (2016). Deep learning for solar power forecasting—an approach using autoencoder and LSTM neural networks. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 002858–002865).
- Giorgino, T. (2009). Computing and visualizing dynamic time warping alignments in r: the dtw package. *Journal of statistical Software*, 31, 1–24.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. (<http://www.deeplearningbook.org>)
- Herve de Beaulieu, M., Jha, M., Garnier, H., & Cerbah, F. (2022). Long range health index estimation based unsupervised RUL prediction using encoder-decoders. In *11th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*. Pafos, Cyprus.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Hu, C., Youn, B. D., Wang, P., & Yoon, J. T. (2012). Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. *Reliability Engineering & System Safety*, 103, 120–135.
- Hu, Y., Palmé, T., & Fink, O. (2016). Deep health indicator extraction: A method based on auto-encoders and extreme learning machines. In *PHM 2016, Denver, USA* (pp. 446–452).
- Jiang, Y., Xia, T., Wang, D., Fang, X., & Xi, L. (2022). Adversarial regressive domain adaptation framework for infrared thermography-based unsupervised remaining useful life prediction. *IEEE Transactions on Industrial Informatics*.
- Khelif, R., Malinowski, S., Chebel-Morello, B., & Zerhouni, N. (2014). Unsupervised kernel regression modeling approach for RUL prediction. In *PHM Society European Conference* (Vol. 2).
- LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*, 3361(10), 1995.
- Lee, J., Wu, F., Zhao, W., Ghaffari, M., Liao, L., & Siegel, D. (2014). Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. *Mechanical systems and signal processing*, 42(1-2), 314–334.
- Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical systems and signal processing*, 104, 799–834.
- Li, X., Ding, Q., & Sun, J.-Q. (2018). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172, 1–11.
- Malhotra, P., TV, V., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. (2016). Multi-sensor prognostics using an unsupervised health index based on LSTM encoder-decoder. *arXiv preprint arXiv:1608.06154*.
- Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1), 43–49.
- Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *International Conference on Prognostics and Health Management* (pp. 1–9). Denver, Colorado, USA.
- Si, X.-S., Wang, W., Hu, C.-H., & Zhou, D.-H. (2011). Remaining useful life estimation—a review on the statistical data driven approaches. *European Journal of Operational Research*, 213(1), 1–14.
- Wang, T., Yu, J., Siegel, D., & Lee, J. (2008). A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *International Conference on Prognostics and Health Management* (pp. 1–6). Denver, Colorado, USA.
- Wu, Y., Yuan, M., Dong, S., Lin, L., & Liu, Y. (2018). Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing*, 275, 167–179.
- Yu, W., Kim, I. Y., & Mechefske, C. (2019). Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme. *Mechanical Systems and Signal Processing*, 129, 764–780.
- Zhang, H., Zhang, Q., Shao, S., Niu, T., & Yang, X. (2020). Attention-based LSTM network for rotatory machine remaining useful life prediction. *IEEE Access*, 8, 132188–132199.

APPENDIX

A. SUMMARY OF THE SELECTED SENSORS OF FD001 DATASET

Sensor ID	Description	Unit
s2	Total temperature at LPC outlet	°R
s3	Total temperature at HPC outlet	°R
s4	Total temperature at LPT outlet	°R
s7	Total pressure at HPC outlet	psia
s11	Static pressure at HPC outlet	psia
s12	Ratio of fuel flow to Ps30	-
s15	Bypass Ratio	-

Autoencoder based Anomaly Detection and Explained Fault Localization in Industrial Cooling Systems

Stephanie Holly¹, Robin Heel², Denis Katic³, Leopold Schoeffl⁴, Andreas Stiftinger⁵, Peter Holzner⁶, Thomas Kaufmann⁷, Bernhard Haslhofer⁸, Daniel Schall⁹, Clemens Heitzinger¹⁰, and Jana Kemnitz¹¹

^{1,2,6,7,9,11} *Siemens Technology, Vienna, 1210, Austria*

stephanie.holly@siemens.com

robin.heel@siemens.com

peter.holzner@siemens.com

thomas.kaufmann@siemens.com

daniel.schall@siemens.com

jana.kemnitz@siemens.com

^{1,6,7,10} *Vienna University of Technology, Vienna, 1040, Austria*

clemens.heitzinger@tuwien.ac.at

^{3,8} *Austrian Institute of Technology, Vienna, 1210, Austria*

denis.katic@ait.com

bernhard.haslhofer@ait.com

^{4,5} *Hauser, Linz, 4040, Austria*

leopold.schoeffl@hauser.com

andreas.stiftinger@hauser.com

ABSTRACT

Anomaly detection in large industrial cooling systems is very challenging due to the high data dimensionality, inconsistent sensor recordings, and lack of labels. The state of the art for automated anomaly detection in these systems typically relies on expert knowledge and thresholds. However, data is viewed isolated and complex, multivariate relationships are neglected. In this work, we present an autoencoder based end-to-end workflow for anomaly detection suitable for multivariate time series data in large industrial cooling systems, including explained fault localization and root cause analysis based on expert knowledge. We identify system failures using a threshold on the total reconstruction error (autoencoder reconstruction error including all sensor signals). For fault localization, we compute the individual reconstruction error (autoencoder reconstruction error for each sensor signal) allowing us to identify the signals that contribute most to the total reconstruction error. Expert knowledge is provided via look-up table enabling root-cause analysis and assignment to

the affected subsystem. We demonstrated our findings in a cooling system unit including 34 sensors over a 8-months' time period using 4-fold cross validation approaches and automatically created labels based on thresholds provided by domain experts. Using 4-fold cross validation, we reached a F1-score of 0.56, whereas the autoencoder results showed a higher consistency score (CS of 0.92) compared to the automatically created labels (CS of 0.62) – indicating that the anomaly is recognized in a very stable manner. The automatically created labels, however, detected anomaly earlier. The main anomaly was found by the autoencoder and automatically created labels, and was also recorded in the log files. Further, the explained fault localization highlighted the most affected component for the main anomaly in a very consistent manner.

1. INTRODUCTION

Malfunctions or even a failure of refrigeration systems are a risk with very high damage potential for food wholesalers. In the course of Industry 4.0 and digitization, sensors and instrumentation drive the central forces of innovation. New potentials for monitoring and machine learning based predictive maintenance of the cold stores are opening up. However,

Stephanie Holly et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

the training and updating of such machine learning based models poses several challenges. First, damage and outage reports, which represent the required ground-truth for a supervised learning task, are not yet collected in a systematic and consistent manner. Second, a refrigerating system is a large, complex system including several hundreds of sensors presenting a widely varying data domain such as temperature, vibration or engine speed (Weerakody, Wong, Wang, & Wendell, 2021). Third, the sensors are often retrofitted or upgraded successively in the course of digitization. Therefore the question remains open how a machine learning model can be scaled from one component to an entire system or several systems.

The challenge of missing ground truth data and therefore learning useful representations with little or no supervision is a key challenge in machine learning. In the context of predictive maintenance, unsupervised learning has shown to be successful in identifying system failures without supervision. However, identifying the root cause of the failure employing unsupervised learning procedures remains an open task. For this reason we transform our unsolvable task into a closely related solvable task, which aims to achieve a similar benefit and business impact. Therefore, we have started with a requirements analysis. The outcomes of the requirements analysis showed us that the localization of the affected sensors, associated with the root cause of the detected failures are an important step. As the affected system is very complex including several hundreds of sensors, the maintenance employee can save valuable time and effort if the affected sensors and thereby the affected subsystem can be localized. Therefore we propose an algorithm for fault localization in large cooling systems with no supervision or little supervision.

- C1:** We define a real-world learning task based on industrial requirements and provide a 18 months ground truth data set for an entire cooling system unit including 34 sensor signals.
- C2:** We provide an autoencoder based anomaly detection workflow suitable for multivariate and increasingly upgraded time series data.
- C3:** Our workflow includes an algorithm for explained fault localization based on the individual reconstruction error for each sensor signal.
- C4:** Our workflow includes a root cause analysis enabled by integrated expert knowledge.
- C5:** Our workflow was compared against automatically created labels showing an F1-score of 0.56 and a consistency score of 0.92. The explained fault localization highlighted the most affected component in a very consistent manner.

2. RELATED WORK

Industrial cooling systems (ICS) are widely deployed in large supermarkets and storage warehouses to preserve per-

ishables with a global market valued to over USD 5 billion (Reportlinker, 2020). ICS are subject to faults, such as compressor failures and bearing damage, that can degrade the operational efficiency and even result in their breakdown. Accurate and timely detection of faults and degradation is critical to prevent food spoilage, customer inconvenience, maintenance costs, and other related losses. Automated fault diagnosis in ICS has been explored for many years (Grimmelius, Klein Woud, & Been, 1995), ranging from Kalman-filter based methods (Yang, Rasmussen, Kieu, & Izadi-Zamanabadi, 2011), random forest (Kulkarni, Devi, Sirighee, Hazra, & Rao, 2018) and neural network based approach. AI based predictive maintenance is estimated to decrease breakdowns by up to 70 % and lowers maintenance costs by 25 % (Deloitte, 2017) in the next years.

Prognostics and health management approaches have been studied extensively across industrial applications, such as aircraft systems (Bieber, Verhagen, & Santos, 2021), hard drives (Barelli & Ottaviani, 2021). While some traditional methods for fault detection require feature engineering (Kato, Yairi, & Hori, 2001; Su, Sun, Gao, Qiu, & Tian, 2019; J. Wang et al., 2020), recent work has shown that end-to-end autoencoders can outperform traditional approaches (Zong et al., 2018; Maleki, Maleki, & Jennings, 2021; Heistracher, Jalali, Suendermann, et al., 2021). Autoencoders are promising and have been for minimal-configuration fault detection (Heistracher, Jalali, Suendermann, et al., 2021; Hood et al., 2021), however this unsupervised methods lack the ability for fault classification or fault location.

Fault classification is typically based on supervised learning (Ismail Fawaz, Forestier, Weber, Idoumghar, & Muller, 2019; Dempster, Petitjean, & Webb, 2020; Z. Wang, Yan, & Oates, 2017; Karim, Majumdar, Darabi, & Chen, 2018) and required a large number of ground truth data, often lacking in industrial applications. Furthermore, these supervised approaches are often difficult to scale and difficult to transfer to other, similar systems (Kemnitz, Bierweiler, Grieb, von Dosky, & Schall, 2021; Heistracher, Jalali, Strobl, et al., 2021). As the affected system is very complex including several hundreds of sensors, the maintenance employee can save valuable time and effort if the affected sensors and thereby the affected subsystem can be localized. Further, explaining and quantifying the individual contribution helps increase trust and interpretability (Grezmak, Wang, Sun, & Gao, 2019). The knowledge about individual localization and contribution can be combined with expert knowledge and thereby enables root cause analysis.

Therefore we propose an end-to-end autoencoder based workflow for fault localization in large cooling systems with no supervision or little supervision. This proposed workflow is inspired by heat and saliency maps (Simonyan, Vedaldi, & Zisserman, 2014) applied in computer vision (Goebel et al., 2018).

3. INDUSTRIAL REQUIREMENTS

Hauser aims to build a machine learning based monitoring, alarm, and remote maintenance systems for industrial refrigeration systems, which are deployed in hundreds of locations around the world. Since many of these systems are of the same type, a model-driven approach that could predict damages or outages to cold storage’s, would scale from a business perspective and could also be offered as a service to customers. The following requirements result from this vision:

- R1: Employable without ground truth data
- R2: Scalable to similar refrigeration systems
- R3: A holistic approach employing a cockpit view
- R4: Employable for fault detection
- R5: Employable for fault localization
- R6: Data agnostic
- R7: Dealing with retrofitted or upgraded sensors

It is important to start with an approach that is expandable. Each step should add substantial business value.

4. MACHINE LEARNING WORKFLOW

4.1. Overall Workflow

We propose a workflow for preprocessing, anomaly detection, explained fault localization and root cause identification of multivariate time series data, see Fig. 1. For anomaly detection, we use a LSTM autoencoder. We identify system failures using a threshold on the total reconstruction error $RE_{total}(S)$ Eq. (7) of sensor signals $S := [S_1, \dots, S_n]$. For fault localization, we compute the individual reconstruction error $RE_{ind}(S_i)$ Eq. (9) for each sensor signal S_i allowing us to identify the signals that contribute the most to the total reconstruction error $RE_{total}(S)$. Provided an expert knowledge "look-up table", we can thus locate the affected subsystem in the cooling system.

4.2. Preprocessing

Data preprocessing is a crucial task in machine learning pipelines (Leukel, González, & Riekert, 2021). Here, data preprocessing denotes the process of preparing raw data for the machine learning model including data cleaning, signal selection, resampling, missing values treatment and data normalization. In industrial systems, data preprocessing is a major challenge due to its high dimensionality and complex structure (Bekar, Nyqvist, & Skoogh, 2020). In industrial cooling systems, we find numerous components with various data sources including several hundreds of sensors, a widely varying data domain, and retrofitted and upgraded sensors requiring distinct approaches in the mentioned preprocessing steps.

In the following, we want to present our approach to data preprocessing in an industrial cooling system with time series

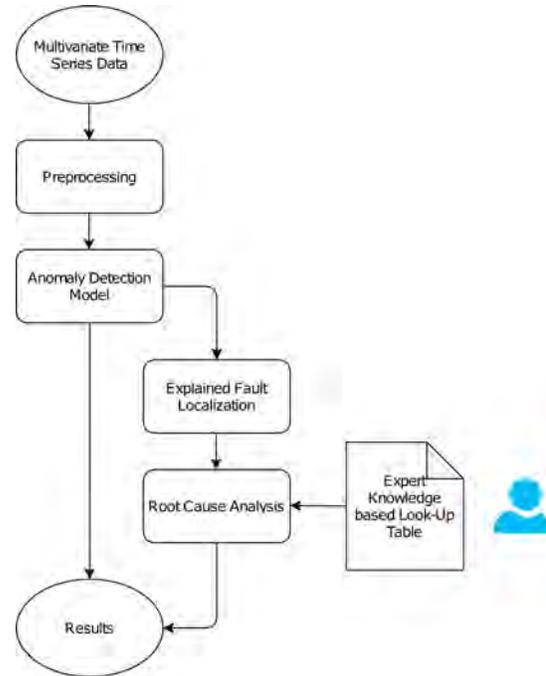


Figure 1. Overall workflow: preprocessing, anomaly detection, explained fault localization, and root cause analysis.

data. Let S_i denote the i^{th} sensor signal in the cooling system, $1 \leq i \leq n$. The data is acquired without a defined frequency, that is, the data is not available at the same frequency and each sensor signal S_i is recorded with an individual frequency or irregularly.

4.2.1. Data Cleaning

In the course of digitization, sensors are often retrofitted or upgraded successively. We therefore removed early time periods lacking sensor signals considered important by Hauser experts.

4.2.2. Signal Selection

The industrial cooling system consists of numerous components with a wealth of data sources providing a huge amount of sensor signals. The massive amount of data is a key challenge in data preprocessing (Bekar et al., 2020). In order to decrease the number of signals, we calculated the correlation coefficient and removed highly correlated signals (Nahian et al., 2021). Let X and Y denote random variables with covariance $cov(X, Y)$ and standard deviation σ_X, σ_Y respectively. Then, the correlation coefficient $\rho_{X,Y}$ is given by

$$\rho_{X,Y} := \frac{cov(X, Y)}{\sigma_X \sigma_Y}. \quad (1)$$

Extracting information from the timestamp can increase the quality of prediction models (Latyshev, 2018). Thus, we

added the signals month, time (hour) and weekday to our observations.

4.2.3. Resampling

The acquired data consists of irregularly sampled time series data. With the growth of multi-sensor systems, the preprocessing of irregular time series data is becoming increasingly important (Weerakody et al., 2021). Due to numerous components with various data sources including several hundreds of sensors and a widely varying data domain, we cannot expect to find all sensor signals sampled at a constant sampling rate with common timestamps. For modeling, however, we need to resample the data to a regular frequency. Let x_t^i denote an observed value of sensor signal S_i at time t . Let r denote a regular sampling rate. We then obtain new timestamps \mathcal{T} by equidistant time intervals of length r . Depending on the signal type, we compute a new value \hat{x}_t^i of signal S_i at timestamp $t \in \mathcal{T}$. In general, the acquired sensor signals represent numerical values, e.g. measuring temperature, vibration or engine speed. However, the cooling system also includes signals of boolean values giving information about the system's health and up-counting signals giving information about the pause time of components in the system. When the value is counting, the minimum value is the most representative one, as it was the initial position of the counter. And using the maximum boolean value, was the same result as using the logical OR, which means that it is zero if and only if all samples of the corresponding interval are zero. In the general case, we simply average over all observed values x_u^i in the interval $[t, t+r)$, that is,

$$\hat{x}_t^i := \frac{1}{\#\{x_u^i | u \in [t, t+r)\}} \sum_{u \in [t, t+r)} x_u^i. \quad (2)$$

In the case of boolean values $x^i \in \{0, 1\}$, we define

$$\hat{x}_t^i := \max_{u \in [t, t+r)} x_u^i, \quad (3)$$

and in the case of constantly (by $c \in \mathbb{N}$) increasing values $x^i \in \{k+c | k \in \mathbb{N}\}$, we define

$$\hat{x}_t^i := \min_{u \in [t, t+r)} x_u^i. \quad (4)$$

4.2.4. Missing Values Treatment

Resampling irregularly sampled time series data will result in missing values for one or more sensor signals S_i at a given timestamp $t \in \mathcal{T}$. Simple statistical techniques include forward-filling and zero imputation (Weerakody et al., 2021). We used the fill-forward method for missing values. Before the first appearance of a value, we initialized a default value by computing the median of all available values.

4.2.5. Feature Representation

In contrast to the acquired signals in the cooling system, let features define the representation of data feed into the machine learning model. Let w denote the window size. Applying a window of size w to the sensor signals S_i at timestamp t , the features F^t at timestamp t are given by

$$F^t := \begin{pmatrix} \hat{x}_t^1 & \hat{x}_t^2 & \dots & \hat{x}_t^n \\ \hat{x}_{t+1}^1 & \hat{x}_{t+1}^2 & \dots & \hat{x}_{t+1}^n \\ \vdots & \vdots & & \vdots \\ \hat{x}_{t+w-1}^1 & \hat{x}_{t+w-1}^2 & \dots & \hat{x}_{t+w-1}^n \end{pmatrix} \in \mathbb{R}^{w \times n}. \quad (5)$$

For modeling, we reshaped the features and obtained

$$F^t = \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_{wn} \end{pmatrix} \in \mathbb{R}^{wn}, \quad (6)$$

where F_j with $j = iw + k$ corresponds to \hat{x}_{t+k}^i .

4.2.6. Data Normalization

Using min-max normalization, we standardized the features by scaling each feature individually to the range $(0, 1)$. The training data is scaled, and then the scaling parameters are applied to the test data. In the domain of machine learning, normalization plays a key role in the preprocessing of data including variables of different scale. In normalization, each variable is scaled individually to the range $(0, 1)$ avoiding a variable dominating the machine learning model.

4.3. Anomaly Detection Model

We applied a LSTM autoencoder with the following settings: an input layer of size 1×370 (37 signals and a window of size 10), the first encoding layer with output size 1×370 , the second encoding layer with output size 1×185 , a repeat vector with output size 1×185 , the first decoding layer with output size 185, the second decoding layer with output size 370, and a time distributed layer of size 1×370 , see Fig. 2.

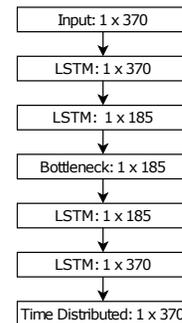
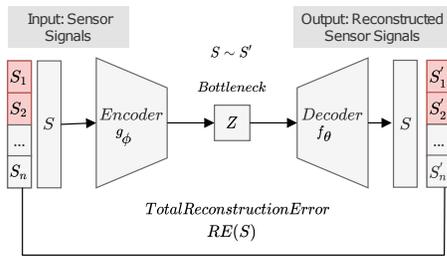


Figure 2. Model Architecture

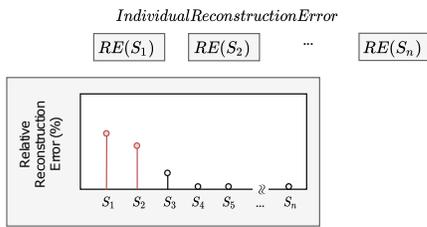
For hyper-parameter tuning, we used Bayesian optimization. Turner et al. (Turner et al., 2020) demonstrated decisively the benefits of Bayesian optimization over random search and grid search for tuning hyperparameters of machine learning models. Bayesian optimization benefits from previous evaluations of hyper-parameter configurations by including past hyper-parameter configurations in the decision of choosing the next hyper-parameter configuration. Therefore, it avoids unnecessary evaluations of the expensive objective function and requires fewer iterations to find the best hyper-parameter configuration.

We searched for the sampling rate, number of layers, dropout rate, activation function, optimizer, learning rate and batch size. We obtained a sampling rate of 60 seconds, 2 autoencoder layers, a dropout rate of 0.2, the tanh activation function, the rmsprop optimizer, a learning rate of 0.001, and a batch size of 16.

① Anomaly Detection Model



② Explained Fault Localization



③ Root Cause Analysis

Expert Knowledgebased Look - Up Table

Sensor	Component	Failure Type
Sensor 1	Component 1	1
Sensor 2	Component 1	1
Sensor 3	Component 2	2

← Domain Expert

Figure 3. Graphical Abstract of Paper, including 1) anomaly detection (identify system failures using a threshold on the total reconstruction error of all sensor signals), 2) explained fault localization (compute the the individual reconstruction error for each sensor signal and identify affected signals), 3) and root cause analysis (locate the affected subsystem based on a look-up table and the affected signals).

4.4. Explained Fault Localization

We propose an end-to-end autoencoder based workflow for fault localization in a large cooling system not providing ground truth data. The total reconstruction error is given by Eq. (7). Motivated by heat and saliency maps (Simonyan et al., 2014) applied in computer vision (Goebel et al., 2018), we derive from the individual reconstruction error Eq. (9) how much each sensor signal S_i contributes to the total reconstruction error Eq. (7). Thus the individual reconstruction error Eq. (9) can be used to identify affected signals and thereby locate the affected subsystem.

The total reconstruction error of feature F^t at timestamp t is given by

$$RE_{total}(F^t) := \frac{1}{wn} \sum_{j=1}^{wn} (F_j - F'_j)^2, \quad (7)$$

where

$$F'^t := \begin{pmatrix} F'_1 \\ F'_2 \\ \vdots \\ F'_n \end{pmatrix} \in \mathbb{R}^{wn} \quad (8)$$

is the prediction of feature F^t at timestamp t .

We define the individual reconstruction error of signal S_i , $1 \leq i \leq n$, at timestamp t by

$$RE_{ind}(S_i) := \frac{1}{w} \sum_{k=0}^{w-1} (\hat{x}_{t+k}^i - \hat{x}'_{t+k}{}^i)^2, \quad (9)$$

where \hat{x}_{t+k}^i and $\hat{x}'_{t+k}{}^i$ correspond to F_j and F'_j with $j = iw+k$ respectively.

In a k -fold cross validation approach, we compute for each test dataset l , $1 \leq l \leq k$, a threshold

$$T_l := \mu_l + c\sigma_l, \quad (10)$$

where μ_l and σ_l are the mean and the standard deviation of the total reconstruction error RE_{total} on the remaining training data and c is a constant found by plotting ROC curves, see Fig. 5.

In algorithm 1, we give the pseudocode for the Fault Localization algorithm. The algorithm takes the threshold T , the feature F^t at timestamp t , the predicted feature F'^t at timestamp t and the number of significant signals m , $1 \leq m \leq n$, as an argument, and returns either the m most significant signals S^* or the empty set. We compute the total reconstruction error RE_{total} of feature F^t at timestamp t based on Eq. (7). If the total reconstruction error RE_{total} exceeds the threshold T , we determine the m most significant signals

$S^* \subset \{S_1, S_2, \dots, S_n\}$. Therefore, we compute the individual reconstruction error $RE_{\text{ind}}[i]$ based on Eq. (9), $i = 1 \dots n$. Then, we select iteratively the signal idx that yields the highest individual reconstruction error $RE_{\text{ind}}[idx]$, append signal idx to S^* and remove the selected signal from further calculations by setting its value to zero. The method APPEND takes a set and an element as an argument and appends the element to the set. When the iteration terminates, set S^* contains exactly the m most significant signals, that is, the signals with the highest individual reconstruction error. If the total reconstruction error RE_{total} does not exceed the threshold T , S^* is the empty set. Finally, the set S^* is returned.

Algorithm 1: Fault Localization

Input: threshold T , feature F^t at timestamp t , predicted feature F'^t at timestamp t , number of significant signals m
Output: significant signals $S^* \subset \{S_1^1, S_1^2, \dots, S_1^n\}$ with $\#S^* = m$
 $RE_{\text{total}} := \frac{1}{wn} \sum_{j=1}^{wn} (F_j - F'_j)^2$
if $RE_{\text{total}} > T$ **then**
 $RE_{\text{ind}}[i] := \frac{1}{w} \sum_{k=0}^{w-1} (\hat{x}_{t+k}^i - \hat{x}'_{t+k}^i)^2 \quad i = 1, \dots, n$
 $S^* := \emptyset$
 for $i = 1, \dots, m$ **do**
 $idx := \arg \max_{i=1, \dots, n} RE_{\text{ind}}[i]$
 APPEND(S^*, idx)
 $RE_{\text{ind}}[idx] := 0$
 end
else
 $S^* := \emptyset$
end
 return S^*

4.5. Root Cause Analysis and Integrated Expert Knowledge

The proposed workflow for fault detection and root cause identification – the identification of the affected sensor signals S_i – allows us to locate the affected subsystem. Each sensor signal S_i is assigned to a component in the cooling system. The component can then be used to determine the root-cause and failure type. Domain knowledge and physical connections are stored in the system using a look-up table. Over the years, the experts have collected which sensors are typically associated with a root-cause. In our case, a physics-aware look up table, see Fig. 3, was created by three domain experts with several years of maintenance experience. While all previous steps in the workflow are fully automatic - and salable to other components - the look-up table will remain component specific and will always require a manual step.

5. EVALUATION

5.1. Data Set Description

The acquired data consists of 34 irregularly sampled sensor signals in an industrial cooling system including numerous

components with various data sources and a widely varying data domain, measured over a period of 18 months. The cooling system is divided into numerous components, and each of them then again divided into several sub-components. Each sensor signal is assigned to a sub-component in the cooling system. We added 3 additional timestamp signals including month, time (hour) and weekday to our observations resulting in a total of 37 signals. By resampling the data to a regular frequency of 60 seconds, we obtained a time series dataset of 37-dimensional data samples. In general, the acquired sensor signals represent numerical values, e.g. measuring temperature, vibration or engine speed. However, the cooling system also includes signals of boolean values giving information about the system's health and up-counting signals giving information about the pause time of components in the system.

In the course of digitization, sensors are often retrofitted or upgraded successively and additional sensors are integrated in the system posing challenges in the preprocessing and updating of the machine learning model. In general, the number of measured data points greatly increases over the measurement period of 18 months for each sensor signal. For missing signals, we computed the mean and standard deviation and sampled from the corresponding normal distribution.

The dataset will be made public available, however, it will be anonymized due to privacy reasons.

5.2. Experimental Setup

Due to the steady increase in the number of data points in the system over 18 months, data acquired in the first 10 months is not representative and thus inadequate for testing. Therefore, we restricted our test data to the last 8 months. We evaluated the machine learning workflow in two different cross validation approaches. In scenario 1, we performed a conventional 4-fold cross validation procedure on the last 8 months (datasets 7–10). In scenario 2, we performed a 4-fold cross validation approach on the last 8 months (datasets 7–10), including a basic training dataset consisting of the first 10 months. We partitioned the data of the last 8 months into 4 equally sized folds, each fold receiving 54 days of acquired data. For each fold k , we performed the following steps: Fold k is held out for testing, and the remaining 3 folds are used for training, in scenario 2 the training data includes the basic dataset. The training data is preprocessed and prepared for modeling using the preprocessing steps described in section 4.2. We then train the LSTM autoencoder described in section 4.3 for 50 epochs with early-stopping on the training data. Finally, we test the autoencoder on the held-out test data.

Table 1. Organization of training and testing dataset in scenario 1 and 2 where 0 refers to the training dataset and 1 refers to the testing dataset

round	scenario 1				scenario 2			
	1	2	3	4	1	2	3	4
dataset 1					0	0	0	0
dataset 2					0	0	0	0
dataset 3					0	0	0	0
dataset 4					0	0	0	0
dataset 5					0	0	0	0
dataset 6					0	0	0	0
dataset 7	0	0	0	1	0	0	0	1
dataset 8	0	0	1	0	0	0	1	0
dataset 9	0	1	0	0	0	1	0	0
dataset 10	1	0	0	0	1	0	0	0

5.3. Ground Truth provided by Automatically Created Labels

We tested our proposed workflow for anomaly detection, fault localization and root cause analysis with thresholds based on expert knowledge, derived from PLC-system, failure log files, system sheets and documentation. A programmable logic controller (PLC) is a device used to control a machine or industrial system. The thresholds have been repeatedly evaluated in extensive feedback discussions with several domain experts and statistically confirmed. In case of absence of expert knowledge, we derived thresholds based on the 98% confidence interval. We obtained for each signal a threshold allowing us to provide automatically created labels. In the preprocessing procedure, after data cleaning, signal selection, resampling, missing values treatment but before data normalization, we compared the input features with the derived thresholds and obtained for each timestamp and signal a label (healthy – 0, anomalous – 1) enabling us to define a label for each sample

$$F^t = \begin{pmatrix} \hat{x}_t^1 & \hat{x}_t^2 & \dots & \hat{x}_t^{37} \\ \hat{x}_{t+1}^1 & \hat{x}_{t+1}^2 & \dots & \hat{x}_{t+1}^{37} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{x}_{t+9}^1 & \hat{x}_{t+9}^2 & \dots & \hat{x}_{t+9}^{37} \end{pmatrix} \in \mathbb{R}^{10 \times 37}. \quad (11)$$

We called a timestamp $t + j$, $0 \leq j \leq 9$, anomalous if at least 10 signals of all 37 signals were anomalous at that timestamp, that is, at least 10 of all values $\hat{x}_{t+j}^1, \hat{x}_{t+j}^2, \dots, \hat{x}_{t+j}^{37}$ exceed their thresholds. Finally, we called a sample F_t with window size $w := 10$ anomalous if $t + j$ was an anomalous timestamp for all $0 \leq j \leq 9$. Then, we applied a smoothing filter to the labels, and labelled a sample F_t anomalous if the smoothed value was greater or equal to 0.5. Fig. 4 shows the labels of data over a time period of 8 months (datasets 7–10 in Fig. 7 and Fig. 8) and compares the results of the corresponding models to the automatically created labels (ground truth).

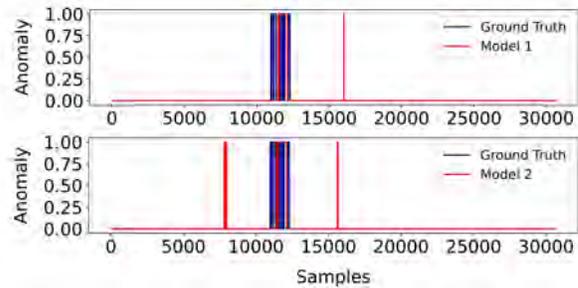


Figure 4. Comparison of anomalies (healthy – 0, anomalous – 1) defined by ground truth or found by model over a time period of 8 months, with models trained in scenario 1 and 2

5.4. Evaluation Metrics

In order to validate our model performance, we computed evaluation metrics including the F1 score, precision and recall for both scenarios, see Table 2. Further, we included ROC curves plotting the false-positive rate against the true-positive rate in several threshold settings, see Fig. 5. The true-positive rate is also known as sensitivity or recall. The ROC curves also helped us find the thresholds Eq. (10). Inspired by the imaging domain, we used the Jaccard index to measure the similarity of machine learning and threshold derived anomalies. The Jaccard index measures the similarity of two datasets comparing their intersection and union.

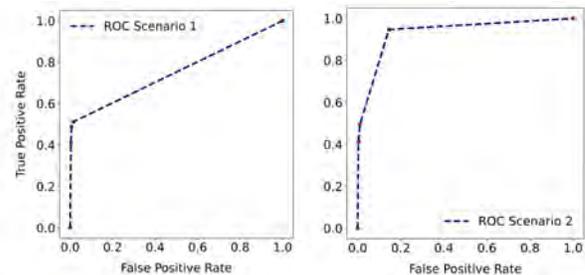


Figure 5. ROC Curves based on 3 different thresholds for each scenario respectively

Table 2. Evaluation Metrics of Scenario 1 and Scenario 2

	Scenario 1	Scenario 2	Ground Truth
f1 score	0.562	0.527	
precision	0.661	0.564	
recall	0.489	0.495	
jaccard index	0.391	0.358	
consistency score	0.920	0.773	0.619

Fig. 6 shows a part of Fig. 4. We use automatically created labels as ground truth references, however, these labels are also affected by precision and recall errors. We observe that the labels are not consistent in the occurrence of an anomaly. In order to measure the reliability of the system, we define a

consistency score indicating the consistency of an anomaly over time, see Table 2. It is based on the assumption that, in the real-world, errors often persist consistently over a longer period of time.

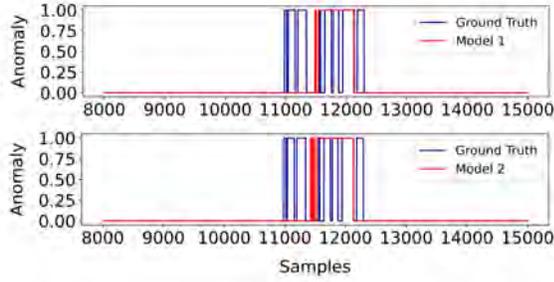


Figure 6. Comparison of an anomaly (healthy – 0, anomalous – 1) defined by ground truth or found by model in dataset 8 (54 days), with a model trained in scenario 1 and 2

Assuming an anomaly a over a time period of several closed intervals $[s_1^a, e_1^a], [s_2^a, e_2^a], \dots, [s_N^a, e_N^a]$, motivated by Fig. 6, we define the consistency score κ_a for an anomaly a as

$$\kappa_a := \frac{1}{e_N^a - s_1^a} \sum_{j=1}^N (e_j^a - s_j^a). \quad (12)$$

Further, we define the consistency score κ for a model as

$$\kappa := \frac{1}{A} \sum_a (e_N^a - s_1^a) \kappa_a \quad (13)$$

where $A := \sum_a (e_N^a - s_1^a)$.

5.5. Experimental Results

We tested our workflow for anomaly detection, fault localization and root cause analysis described in section 4.4 and 4.5 in scenario 1 and scenario 2. Scenario 1 refers to the conventional 4-fold cross validation procedure on the last 8 months (datasets 7–10). Scenario 2 refers to the 4-fold cross validation approach on the last 8 months (datasets 7–10) including a basic training dataset consisting of the first 10 months. Fig. 7 shows the results for anomaly detection in scenario 1. Fig. 8 shows the results for anomaly detection in scenario 2. The plot shows the total reconstruction error Eq. (7) and threshold computed according to Eq. (10) on the respective test dataset over a period of 54 days. Setting the sampling rate $r := 60$ (seconds) and the window size $w := 10$, we obtain 7776 ($6 \times 24 \times 54$) data samples over a period of 54 days. In both scenarios, the autoencoder detected an anomaly in dataset 8 around data sample 12000 and an anomaly in dataset 9 around data sample 16000. However, in scenario 2, the autoencoder, trained on the training data including the basic dataset, detected the anomaly in dataset 9 slightly earlier than in scenario 1. Therefore, we suggest that including the data acquired in the first 10 months

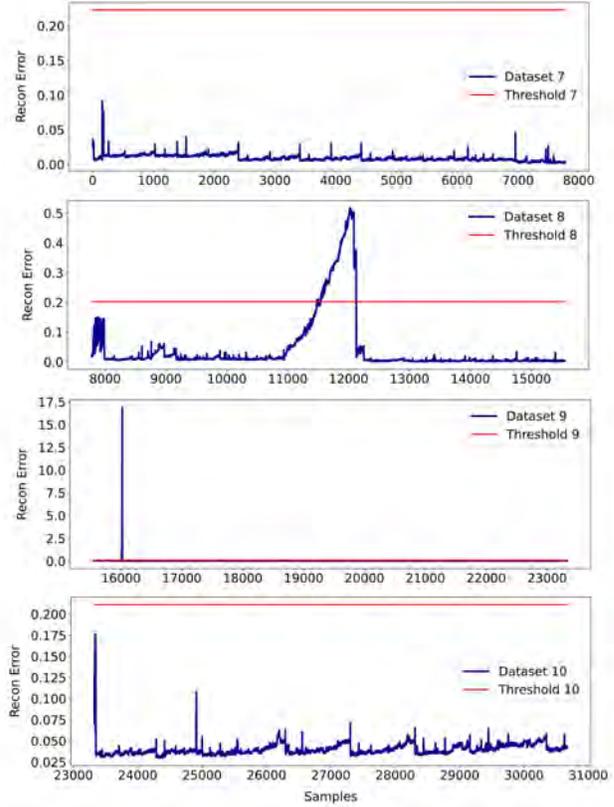


Figure 7. Anomaly Detection in Scenario 1: total reconstruction error on test datasets 7–10, each dataset of 1.8 months (54 days), with respective threshold

increases the performance of the autoencoder.

Fig. 9 shows the results for fault localization in scenario 2 on dataset 8. Fig. 10 shows the results for fault localization in scenario 2 on dataset 9. The plot shows the individual reconstruction error Eq. (9) of an anomalous data sample for all 37 signals, absolutely and relatively with the respective threshold Eq. 10. The color indicates if the individual reconstruction error of signal S_i exceeds the threshold.

Table 3. Root Cause Analysis Scenario 2: component and failure type of anomalous signals of anomalous data sample in dataset 8

RE Contribution (%)	Sensor	Component	Failure Type
11.94	Sensor 7	Component 3	2
48.86	Sensor 14	Component 1	1
8.91	Sensor 22	Component 1	1
9.33	Sensor 36	Component 13	5

Please note that the data is anonymized due to privacy reasons.

Table 3 shows the results for root cause analysis in scenario 2 on dataset 8. Table 4 shows the results for root cause analysis in scenario 2 on dataset 9. The explained fault localization and root cause analysis highlighted the most affected component

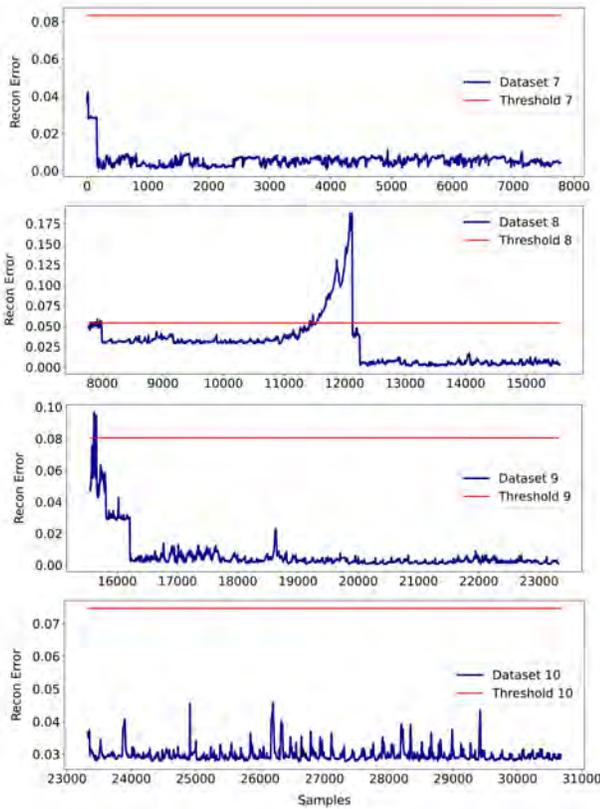


Figure 8. Anomaly Detection in Scenario 2: total reconstruction error on test datasets 7–10, each dataset of 1.8 months (54 days), with respective threshold

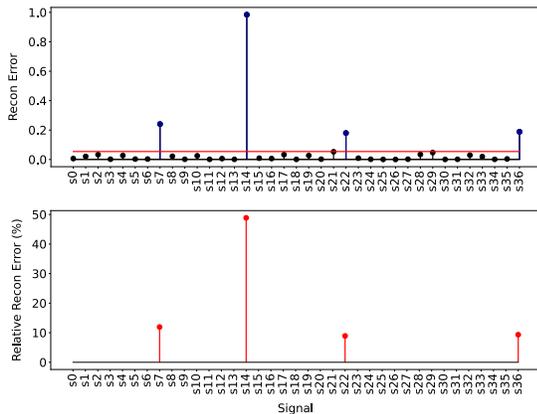


Figure 9. Explained Fault Localization in Scenario 2: individual reconstruction error of anomalous data sample in dataset 8

in a consistent manner.

6. CONCLUSION AND FUTURE WORK

We provided a 18 months dataset of multivariate time series data for an industrial cooling system including 34 sensor signals and automatically created labels based on thresholds de-

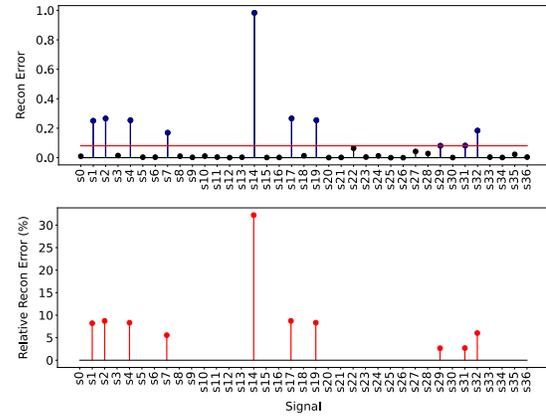


Figure 10. Explained Fault Localization in Scenario 2: individual reconstruction error of anomalous data sample in dataset 9

Table 4. Root Cause Analysis Scenario 2: component and failure type of anomalous signals of anomalous data sample in dataset 9

RE Contribution (%)	Sensor	Component	Failure Type
8.21	Sensor 1	Component 1	1
8.71	Sensor 2	Component 1	1
8.33	Sensor 4	Component 1	1
5.55	Sensor 7	Component 3	2
32.22	Sensor 14	Component 1	1
8.73	Sensor 17	Component 1	1
8.33	Sensor 19	Component 1	1
2.66	Sensor 29	Component 10	3
2.69	Sensor 31	Component 12	3
6.03	Sensor 32	Component 12	3

Please note that the data is anonymized due to privacy reasons.

rived from expert knowledge and PLC-system. We presented our machine learning workflow for anomaly detection and explained fault localization suitable for multivariate and increasingly upgraded time series data. Therefore, we presented our preprocessing steps and provided an algorithm for explained fault localization and a root cause analysis enabled by integrated expert knowledge.

We performed a conventional 4-fold cross validation approach over a time period of 8 months and a 4-fold cross validation approach including a basic dataset of 10 months and compared the model results to automatically created labels based on thresholds provided by domain experts. Using 4-fold cross validation, we reached a F1-score of 0.56, whereas the model results showed a higher consistency score (CS of 0.92) compared to the automatically created labels (CS of 0.62) – indicating that the anomaly is recognized in a very stable manner. The automatically created labels, however, detected anomaly earlier. The main anomaly was found by the model and ground truth, and was also recorded in the log files. Further, the explained fault localization highlighted the most affected component for the main anomaly in a very consistent

manner.

A limitation of this work is the comparison of our model results to automatically created labels as ground truth references. These labels are also affected by precision and recall errors. Still, automatically created labels were the best available reference for this work and are frequently considered ground truth for other real-world applications. However, a strength of our study is that we also created and provided consistency scores indicating the consistency of an anomaly over time for autoencoder results and automatically created labels. We believe that this score will also be helpful in the future for evaluation using real-world data missing ground truth. It is based on the assumption that, in the real-world, errors often persist consistently over a longer period of time.

In the future, we would like to further expand root cause analysis and increase the transparency of our proposed workflow. Further, we aim to investigate whether the increase in reconstruction error and exceeding the anomaly thresholds can be predicted for the user. A further step is to bring the developed end-to-end model into a productive environment.

Our work shows that scalable anomaly detection based on AI and explained fault localization is feasible for multivariate and increasingly upgraded time series data. Our proposed workflow provides a satisfying performance regarding the F1 score. We were also able to show that a root cause analysis enabled by integrated expert knowledge can be carried out without supervised learning highlighting the most affected component in a consistent manner. The integrated expert knowledge enabled ground truth references. We are sure that our results can also be transferred to other applications and will enable the monitoring of large systems in the future.

DATA AVAILABILITY STATEMENT

The dataset will be shared upon reasonable request. Please contact the senior author of the paper. Please cite this paper if you are using the dataset.

ACKNOWLEDGEMENTS

We thank the Hauser experts their for the valuable contribution of domain knowledge, for their time and helpful discussions.

REFERENCES

Barelli, E., & Ottaviani, E. (2021). Unsupervised anomaly detection for hard drives. *Proceedings of the European Conference of the PHM Society 2021*, 6(1), 7. doi: <https://doi.org/10.36001/phme.2021.v6i1.2795>

Bekar, E. T., Nyqvist, P., & Skoogh, A. (2020). An intelligent approach for data pre-processing and analysis in predictive maintenance with an industrial case study. *Advances in Mechanical Engineering*, 12. doi:

10.1177/1687814020919207

Bieber, M., Verhagen, W. J., & Santos, B. F. (2021). An adaptive framework for remaining useful life predictions of aircraft systems. *Proceedings of the European Conference of the PHM Society 2021*, 6(1), 11. doi: <https://doi.org/10.36001/phme.2021.v6i1.2868>

Deloitte. (2017). Predictive maintenance – taking pro-active measures based on advanced data analytics to predict and avoid machine failure. *Deloitte*.

Dempster, A., Petitjean, F., & Webb, G. I. (2020). Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5), 1454–1495. doi: 10.1007/s10618-020-00701-z

Goebel, R., Chander, A., Holzinger, K., Lecue, F., Akata, Z., Stumpf, S., ... Holzinger, A. (2018). Explainable AI: The new 42? *Machine Learning and Knowledge Extraction*, 295–303.

Grezmak, J., Wang, P., Sun, C., & Gao, R. X. (2019). Explainable convolutional neural network for gearbox fault diagnosis. *Procedia CIRP*, 80, 476-481.

Grimmelius, H., Klein Woud, J., & Been, G. (1995). On-line failure diagnosis for compression refrigeration plants. *International Journal of Refrigeration*, 18(1), 31–41. doi: [https://doi.org/10.1016/0140-7007\(94\)P3709-A](https://doi.org/10.1016/0140-7007(94)P3709-A)

Heistracher, C., Jalali, A., Strobl, I., Suendermann, A., Meixner, S., Holly, S., ... Kemnitz, J. (2021). Transfer learning strategies for anomaly detection in IoT vibration data. *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, 1-6. doi: 10.1109/IECON48115.2021.9589185

Heistracher, C., Jalali, A., Suendermann, A., Meixner, S., Schall, D., Haslhofer, B., & Kemnitz, J. (2021). Minimal-configuration anomaly detection for IIoT sensors. *arXiv: 2110.04049*.

Hood, A., Valant, C., Horney, P., Jones, A., Lantner, J. S., Martuscello, J., & Nenadic, N. (2021). Autoencoder based anomaly detector for gear tooth bending fatigue cracks. *Proceedings of the Annual Conference of the PHM Society 2021*, 13, 10.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4), 917–963. doi: 10.1007/s10618-019-00619-1

Karim, F., Majumdar, S., Darabi, H., & Chen, S. (2018). Lstm fully convolutional networks for time series classification. *IEEE Access*, 6, 1662–1669. doi: 10.1109/ACCESS.2017.2779939

Kato, Y., Yairi, T., & Hori, K. (2001). Integrating data mining techniques and design information management for failure prevention. *Annual Conference of the Japanese Society for Artificial Intelligence*, 475–480.

Kemnitz, J., Bierweiler, T., Grieb, H., von Dosky, S., & Schall,

- D. (2021). Towards robust and transferable IIoT sensor based anomaly classification using artificial intelligence. *arXiv:2110.03440*.
- Kulkarni, K., Devi, U., Sirighee, A., Hazra, J., & Rao, P. (2018). Predictive maintenance for supermarket refrigeration systems using only case temperature data. *2018 Annual American Control Conference (ACC)*, 4640–4645. doi: 10.23919/ACC.2018.8431901
- Latyshev, E. (2018). Sensor data preprocessing, feature engineering and equipment remaining lifetime forecasting for predictive maintenance. *DAMDID/RCDL*.
- Leukel, J., González, J., & Riekert, M. (2021). Adoption of machine learning technology for failure prediction in industrial maintenance: A systematic review. *Journal of Manufacturing Systems*, 61, 87–96. doi: 10.1016/j.jmsy.2021.08.012
- Maleki, S., Maleki, S., & Jennings, N. R. (2021). Unsupervised anomaly detection with lstm autoencoders using statistical data-filtering. *Applied Soft Computing*, 108, 107443. doi: <https://doi.org/10.1016/j.asoc.2021.107443>
- Nahian, J. A., Ghosh, T., Banna, H. A., Aseeri, M. A., Uddin, M. N., Ahmed, M. R., ... Kaiser, S. (2021). Towards an accelerometer-based elderly fall detection system using cross-disciplinary time series features. *IEEE Access*, 9, 39413–39431. doi: 10.1109/ACCESS.2021.3056441
- Reportlinker. (2020). Global evaporative cooling market – growth, trends, covid-19 impact, and forecasts (2021 – 2026). *Reportlinker*. Retrieved from <https://www.reportlinker.com/p06179097>
- Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*.
- Su, S., Sun, Y., Gao, X., Qiu, J., & Tian, Z. (2019). A correlation-change based feature selection method for IoT equipment anomaly detection. *Applied Sciences*, 9(3). doi: 10.3390/app9030437
- Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., & Guyon, I. (2020). Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. *NeurIPS 2020 Competition and Demonstration Track*.
- Wang, J., Tang, Y., He, S., Zhao, C., Sharma, P. K., Alfarraj, O., & Tolba, A. (2020). Logevent2vec: Logevent-to-vector based anomaly detection for large-scale logs in internet of things. *Sensors*(9). doi: 10.3390/s20092451
- Wang, Z., Yan, W., & Oates, T. (2017). Time series classification from scratch with deep neural networks: A strong baseline. *2017 International Joint Conference on Neural Networks (IJCNN)*, 1578–1585. doi: 10.1109/IJCNN.2017.7966039
- Weerakody, P., Wong, K. W., Wang, G., & Wendell, E. (2021). A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing*, 441. doi: 10.1016/j.neucom.2021.02.046
- Yang, Z., Rasmussen, K. B., Kieu, A. T., & Izadi-Zamanabadi, R. (2011). Fault detection and isolation for a supermarket refrigeration system – part one: Kalman-filter-based methods. *IFAC Proceedings Volumes*, 44(1), 13233–13238. (18th IFAC World Congress) doi: <https://doi.org/10.3182/20110828-6-IT-1002.03115>
- Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., & Chen, H. (2018). Deep autoencoding gaussian mixture model for unsupervised anomaly detection. *ICLR 2018 Conference*.

Joint Autoencoder-Classifier Model for Malfunction Identification and Classification on Marine Diesel Engine Diagnostics Data

Kürşat İnce^{1,3}, Gazi Koçak², and Yakup Genc³

¹ *Naval Combat Management Technologies Center, HAVELSAN Inc., Pendik/Istanbul, Turkey
kince@havelsan.com.tr*

² *Department of Marine Engineering, Istanbul Technical University, Tuzla/Istanbul, Turkey
kocakga@itu.edu.tr*

³ *Computer Engineering Department, Gebze Technical University, Gebze/Kocaeli, Turkey
kince,yakup.genc@gtu.edu.tr*

ABSTRACT

There has been an increasing demand on marine transportation and traveling, since the voyage of the ships are more economical and efficient than air or land based alternatives. The propulsion of a ship is provided by a main engine system which includes the shaft, the propellers, and other auxiliary equipment. Marine diesel engine is a complex structure that the faults within these machines can cause malfunction of the whole system, which in turn inhibits the ship's mission. It is crucial to monitor the engine and other auxiliary systems during the operation and infer their condition from their diagnostic data. In this study, we analyze monitoring data of a crude oil tanker for different ship loads and conditions. Our primary analysis include main engine fault detection and classification for which we propose an end-to-end joint autoencoder-classifier model that contains a convolutional autoencoder, and a long-short term memory regressor connected to the the latent space. Genetic algorithms optimized models gave us 93.61% accuracy for fault classification task. Further investigation on feature's contributions to the model, we increased the accuracy upto 96%. One concern about marine transportation is the pollution of the air with green house effect gases. In this study, we have developed NOx and SOx emission estimators for different faults and working conditions. Leveraging ship load, working conditions and engine faults in the models helped us to achieve 50% better estimation performance. Although there are other studies regarding gases emissions in the literature, this is the first study that took engine faults into account. We believe that the joint autoencoder-classifier model will be useful for other time series estimation task on other domains, especially

Kürşat İnce et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

where the operating condition plays a role in the process.

1. INTRODUCTION

The propulsion of a ship is provided by a main engine system which includes the shaft, the propellers, and other auxiliary equipment. Almost all the merchant ships utilize marine diesel engines with cylinders, pistons, valves, nozzles, and a turbo charger system, which provides power for the navigation of the ship. The diesel engine is a very complex system that the faults within these machines can cause malfunction of the whole system, which in turn inhibits the ship's mission. So, it is crucial to monitor the engine and other auxiliary systems during the operation and infer their condition from this diagnostic data. The main objective of fault diagnostics are fault detection, fault identification, and fault analysis, which is an essential part of modern industries to ensure safety and product quality. Fault diagnosis has been active area of research for the last few decades (Heo & Lee, 2018).

In this study, we analyze monitoring data of a crude oil tanker at the full ahead loaded and full ahead unloaded situations for main engine fault detection and identification. The data is obtained from a realistic full-mission engine room simulator from Kongsberg. More than 60 sensor data have been recorded for three cases: normal working conditions plus two cases for malfunction scenarios on diesel engine, namely injection valve nozzle wear and injection valve nozzle clogged. Each run of the failure scenarios ends with one of the malfunctions, if any. As part of our initial research on this dataset, we define a classification problem in which we detect existence of any defined malfunctions and identify what the malfunction could be. We have built a joint autoencoder-classifier model, which contains a convolutional autoencoder, and a long-short term memory regressor connected to the the latent space. The joint architecture helps us to train end-to-end

multi-target deep neural network from the multivariate time series data as in the dataset, and to integrate operating condition of the process into the model added to the latent parameters. Using the best performing models, we have investigated the sensors which might alleviate the performance of the classification. The amounts of released NO_x and SO_x emissions were also recorded during data collection process. In this initial study, we also built several gradient boosting based regressors to estimate NO_x and SO_x emissions for different ship loads and for different fault types. Our study shows that a) end-to-end joint autoencoder-classifier model provides up to 95.94% accuracy for classification problems, b) as it is in emission estimation, operating conditions are valuable resource of information for processes, c) leveraging operating conditions and faults in the model helps us to achieve 50% better estimations.

This analysis paper is organized as follows: Section 2 gives information about simulation dataset, which we named MC90-V as the simulator. Section 3 summarizes fault classification problem and gas emissions of the engine. Section 4 describes the workflow and methods we used for analysis. Section 5 describes our implementation details and results.

2. KONGSBERG MC90-V ENGINE ROOM SIMULATOR DATASET

Kongsberg K-Sim is a well-known ship engine room simulator with high fidelity, among maritime departments of the universities. One configuration of the simulator, ERS MAN B&W 5L90MC VLCC L11-V (MC90-V for short), simulates a very large crude carrier with a MAN B&W slow speed turbo charged diesel engine as propulsion unit modeled with fixed and controllable propeller. The model is based on real engine data that make the dynamic behavior of the simulator close to real engine response. The simulator includes control room operator station and panels and bridge and steering panels. K-Sim provides other applications such as Neptune for classroom training and TLDS for engine room monitoring.

We have used Neptune for defining *exercises* that run the core simulator for several times. We have used TLDS to record simulator variables as engine room monitoring data. An exercise in Neptune is defined by an Initial Condition, which is the initial state of the simulator. For the MC90-V dataset, we have created 18 different scenario initializations by defining the conditions on ship’s load and speed, sea water temperature, and sea conditions as given in Table 1.

The simulator provides about 1500 malfunctions to be injected into the exercises. For our research, we restricted ourselves to two of the malfunctions on the 1st cylinder of the engine, namely Cyl 1 injection valve nozzle wear and Cyl 1 injection valve nozzle clogged, which are referred as M2503 and M2508 in this paper. For each initial condition (18 in total) and for each malfunction state (3 in total) we run the

Table 1. Initial Conditions for MC90-V Dataset

Condition	Possible Values
Ship Speed/Load	FAL: Full Ahead Loaded FAU: Full Ahead Unloaded
Sea Water Temperature	20°C 25°C 28°C
Sea Condition (Beauf)	0 4 6

exercises for 53 times, 2862 runs in total. For each initial condition, we separated 35 of the runs for training and the remaining 18 runs for testing. Each run contains 1000 to 1400 data samples recorded at 1 Hz until the failure occurs. The monitoring data contains more than 60 variables, which can be grouped as *real sensors* and *simulated sensors*. For this initial study we have used real sensors.

We present in the paper is the first of many analysis we plan on the MC90-V dataset. For this study we restrict ourselves to ship load *Full Ahead Loaded* and *Full Ahead Unloaded*, sea water temperature 20°C and Sea Condition 0.

3. PROBLEM DEFINITION

Detection of faults in a monitored system is the first step in root cause identification, which is crucial before proceeding to the next stages of the diagnosis process. The main aim of fault detection and diagnosis are to identify key indicators which can be used for health prediction of a system and then to take a proper action against a future failures. This key indicators can be used to predict the fault class before the system losses its operation ability.

MC90-V Dataset provides unique challenges, one of which is *fault identification* and *classification*, i.e. identify the state of the engine, whether it is operating normal, and predicting the fault type if it is not. Fault-free exercise runs, which we labeled as M0000, represent the behavior of the engine during its normal operating regime. In this case, the engine does not present any problem and runs smoothly. Faults regarding the Cyl. 1 are injected by the simulator after a random delay. These faults are labeled as M2503 and M2508. Two of the objectives of this study are to identify and classify the faults in test data and to identify the signals having highest contribution to the prediction.

The main engine of the ships run on diesel fuel that produce several gases while burning. As more and more ships travel in each day, their emissions becomes a global concern. The two main pollutants from the ship’s emission are Nitrogen oxides (NO_x) and Sulphur oxides (SO_x) gases, which effect on the ozone layer in the troposphere area of the earth’s atmosphere and cause the green house effect and global warming. One other objective in this study is to estimate NO_x and SO_x

emissions from the burning process.

4. METHODS AND TECHNIQUES

In this study, our main motivation was to build a fault classifier, which would differentiate between M0000, M2503 and M2508. We have used the workflow as given in Figure 1, which contains data preprocessing, model training, and evaluation phases. Data preprocessing step prepares the data for training and evaluation steps.

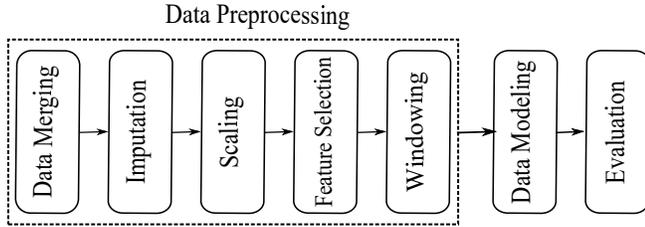


Figure 1. General data processing workflow in this study

Data Preprocessing: The actions we took in this step are data merging, since the simulation data is distributed among 23000 or so files, missing value imputation, data normalization and scaling, feature selection, and windowing. A few of the featured had missing values, which were imputed with *backfill*.

Data Modeling and Optimization: In this study, we propose a joint autoencoder-classifier (JAEC) architecture, which integrates a CNN based autoencoder and an LSTM-based classifier end-to-end for fault classification. The model incorporates CNN based autoencoder. The general architecture is given in Figure 2. Input to autoencoder is $X(t)$, the sensor values at time t . The decoder produces $\hat{X}(t)$, the signals at time t . Encoded sensor values (latent space) for time t is concatenated to operational conditions $OC(t)$ at time t . This data takes the classification path, which contains two LSTM layers and dense layers that generate $\widehat{Class}(t)$. In JAEC-CNN, convolutional layers in the autoencoder are wrapped in time distributed layers, which applies convolution operation to every temporal slice of the input. Number of filters in CNN layers decreases in the encoder to support latent space generation. Batch normalization is performed between convolutional layers.

For NO_x and SO_x emission estimations, we propose gradient boosting (GB) based models. GB is one of the powerful techniques for performing classification and regression tasks that builds the model in a stage-wise fashion. GB is an ensemble learner: a complex model based on a collection of individual models. These individual models may have poor predictive power and are prone to overfitting, but combining many such weak models in an ensemble will lead to a much better outcome overall.

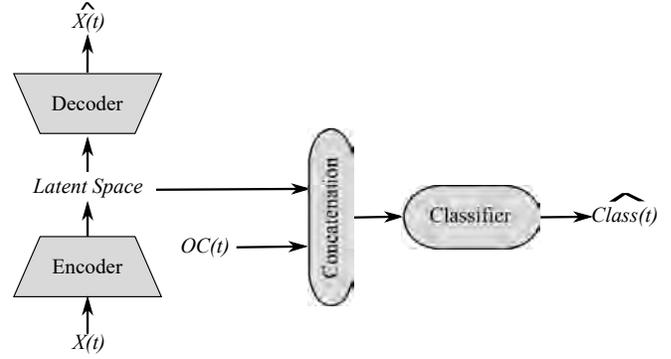


Figure 2. Proposed joint autoencoder classifier architecture

Hyper-parameters for the model architecture are optimized using genetic algorithms (GA), which is based on the biological concept of evolution (Back, Fogel, & Michalewicz, 2000). In genetic algorithms hyper-parameters to optimize are called as *genes*, and a set of gene sequences that construct the unique model is called an *individual*. The genetic algorithms process starts with a set of individuals, i.e. population, which are actually an initial set of models. The initial population is *trained* and *evaluated* for their *fitness* to the problem solution. Usually, n best fitted individuals are selected to form the next generation through gene crossover (mating) and gene mutation. The individuals in the new generation goes into training and evaluation stages. The process continues until maximum number of generations is reached or predetermined fitness score is achieved. Finally, best fitted individuals are selected to create the optimal architectural models.

Evaluation Metrics: Commonly used evaluation metrics for classification problems are accuracy (ACC), and F1 score (F1), which are defined by true positives (TP), true negatives (TN), false positives (FP), false negatives (FN) as in Equation 1 and Equation 2, respectively.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$F1 = 2 \times \frac{TP}{TP + \frac{FP+FN}{2}} \quad (2)$$

Evaluation metrics for regression problems are Mean Absolute Error and Root Mean Square Error are defined as in Equation 3 and Equation 4, respectively. In these equations y is the expected result (i.e. ground truth), \hat{y} is the model estimation, and i is the sample index.

$$MAE(y, \hat{y}) = \left(\frac{1}{n}\right) \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3)$$

$$RMSE(y, \hat{y}) = \sqrt{\left(\frac{1}{n}\right) \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (4)$$

5. EXPERIMENTS AND RESULTS

We have used Scikit-Learn (Pedregosa et al., 2011) and Keras (Chollet et al., 2015) libraries to implement proposed framework for fault classification and emission estimation on MC90-V dataset. This section gives implementation details for each task, and the evaluation results.¹

5.1. Fault Classification

Data preprocessing step prepared the data for training and evaluation steps. Initially, the dataset contains individual data files for each exercises runs. We have merged individual data files to have two separate subsets, training and test. The dataset contained missing data for each run. Because missing data can create problems for analyzing data, interpolation is used to fill in missing data and avoid pitfalls involving cases that have missing values. After that, data normalization is performed using Z-score normalization, with mean zero and standard deviation one. Initially training data is scaled, and then the scaling parameters are applied to test data. Our exploratory data analysis showed that some of the features has zero variance. We removed these features from the dataset before training. Finally we have used window sizes of 5 up to 100 to create a context for the time series data. The rearranged data have passed to model training phase.

Model training and optimization of JAEC model is performed using DEAP Library (Fortin, De Rainville, Gardner, Parizeau, & Gagné, 2012), a Genetic Algorithms (GA) based optimization framework. Hyper-parameters for GA evolutions are given in Table 2. GA optimization parameters are selected manually after a few experimental runs. The hyper-parameters for the best model is retrieved from these optimizations. We have used different percentages of the training and test data starting from the beginning of each run. This was to divide the runs into regions, which in turn helped us to understand how the fault was developing up in each run.

Table 2. Genetic algorithms search parameters for hyper-parameter optimization

GA Parameter	Value
Initial population size	30
Number of generations	7
Population size per generation	10
Mate probability	0.5
Mutation probability	0.5
Number of selected individuals per generation	5

¹The source code for this study will be available at <https://github.com/zakkum42/phme22-public> after the publication of this paper.

5.2. Emission Estimation

We have used the same steps as the fault classification task with the exception that we did not perform windowing. Estimation of NOx and SOx emissions were performed with XG-Boost library (Chen & Guestrin, 2016). Hyper-parameter optimization was also performed. For this task, we were able to use of the fault label for further investigating the gas emissions.

5.3. Results and Discussion

Fault classification results for optimized model are given in Table 3 for different percentages of the training data and for full test data. ACC and F1 metrics are about the same, which is expected as the dataset is well-balanced for each fault. Confusion matrix in Figure 3 shows how ACC is increasing with the addition of new training data. The drop in ACC from 60% to 80% of training data is negligible, and can be attributed to the random initializations of the models. With lesser data M0000 and M2508 were confused the most, but with increasing percentage of the training data the confusion dropped from 21.44% to 2.7%. The other confusions have dropped as well.

Table 3. Classification results for different training percentages

Train (%)	Test (100%)	
	ACC	F1
20	61.78	60.56
40	82.34	81.32
60	91.78	91.78
80	91.14	91.12
100	93.61	93.63

Experimenting with different percentages of the train and test data gave us the scores in Table 4. Highest scores for these experiments are achieved when the same percentage of train and test data was used, i.e. the diagonal of the table. In the left of the diagonal when we added new test samples the scores increased. This was because new samples were coming from known regions as the training data. On the other hand, in the right of the diagonal when we added new test samples the scores decreased. This was because new samples were coming from unknown regions as the training data. One exception to this observation is when the training data was 20% and we increased test samples from 20% to 40% – the first row of the data in Table 4. We suspect that the adjustments to designated initial conditions in the very beginning of the exercise caused fluctuations in the sensor data, whose effect has been dropped with increasing number of samples from a similar region in which the fault has not been fully developed yet.

Using the best classification model, we have analyzed the effect of each feature to the accuracy. Initially we used full train and test data to calculate a base score for the model. Then we iterated over the features: adding noise to a given feature in-

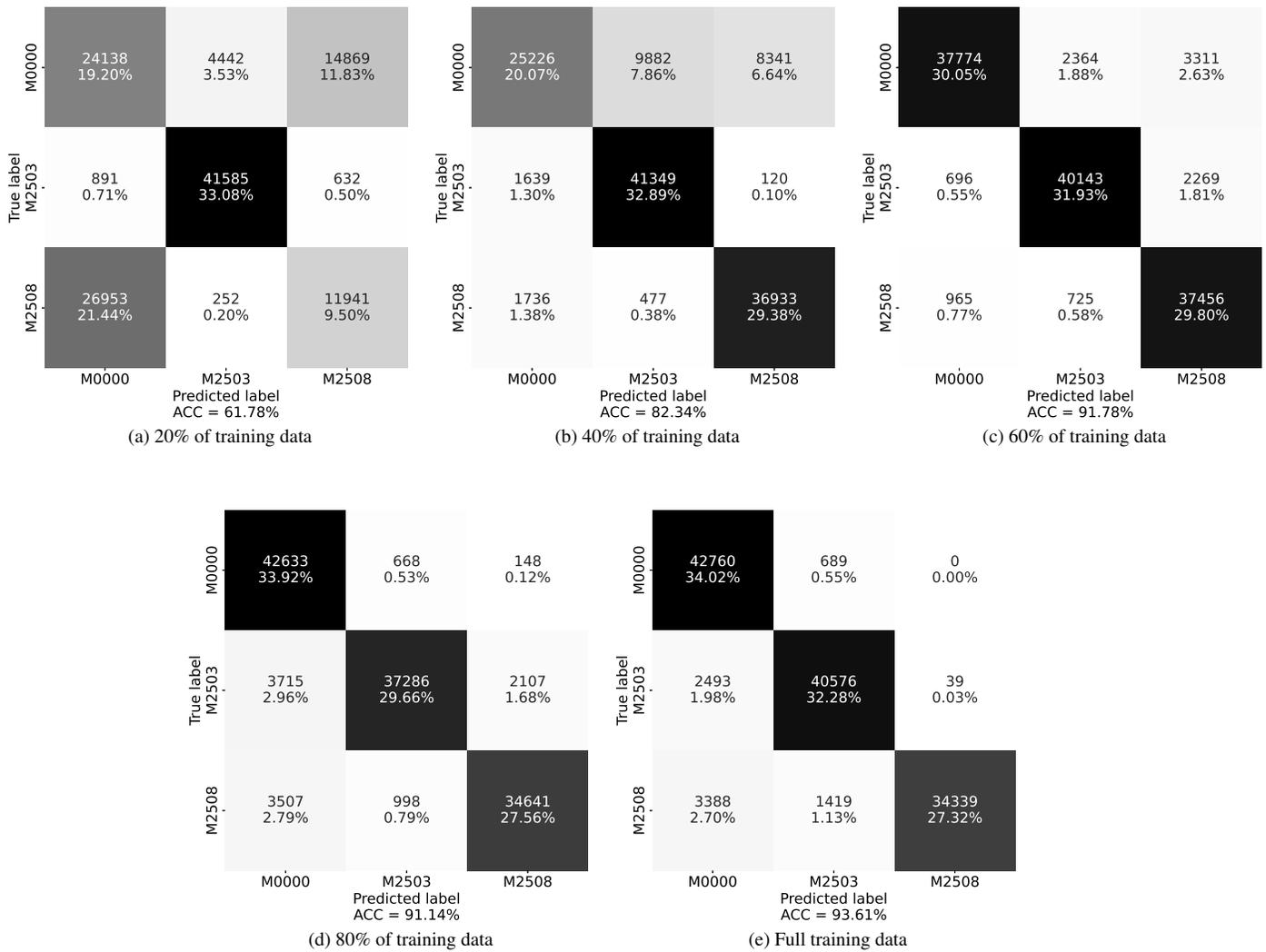


Figure 3. Confusion matrices for varying amount of training data and full test data

Table 4. Classification results for varying amount of train and test data

Test (%)	20		40		60		80		100	
Train (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
20	79.77	79.83	85.39	85.39	73.22	72.88	65.35	64.78	61.78	60.56
40	78.87	79.13	90.55	90.65	90.42	90.45	85.67	85.25	82.34	81.32
60	79.15	79.12	90.68	90.68	94.00	94.00	93.70	93.69	91.78	91.78
80	73.98	74.01	88.37	88.58	92.51	92.63	94.48	94.55	91.14	91.12
100	73.17	73.23	88.01	88.25	92.28	92.41	94.31	94.39	93.61	93.63

validated the features one by one, and a new score can be calculated. The difference between the base score and the new score gave feature’s contribution to the model. Sorted list of contributions gave us the feature rank. The contribution of the features are given in Figure 4. As expected, the top features were directly related to the combustion in the cylinder Cyl 1, such as various temperatures and emissions. Removing features with negative contribution increased ACC from

93.61% to 95.94%, which was 2.5% increase in the overall classification performance.

Emission estimations for NOx and SOx gases are given in Table 5 and Table 6, respectively. When the ship engine did not have any fault, i.e. M0000, the emissions were the lowest as we expected. Smaller FAU and FAL scores suggested that when we knew the ship load, we could have better estimation

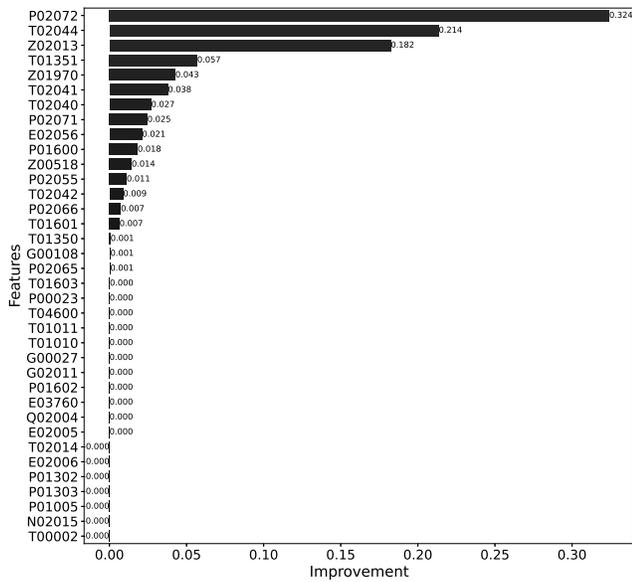


Figure 4. Contribution of each feature to the classification accuracy

of the emissions. When we have used different estimators for ship loads, MAE score decreased upto 50% for full dataset. On the other hand, MAE and RMSE scores for M2508 gas emissions were higher than M2503, which might be an indication of a complications in the burning process that results higher emission variance during M2508 fault development. For comparison, NOx and SOx emissions means and standard deviations of the dataset are given in Table 7 and Table 8, respectively. We observe that our MAE and RMSE scores are comparable to their respective standard deviations.

All the model development and evaluation is done using the simulated dataset. Unfortunately, we could not have any real-world data that we would validate or invalidate our models up-to now.

6. CONCLUSION

In this paper, we presented our initial analysis on MC90-V dataset, which was constructed via Kongsberg K-Sim, a well-known ship engine room simulator. For this study we aimed for fault identification and classification, namely Cyl 1 injection valve nozzle wear and Cyl 1 injection valve nozzle clogged. For classification, we used a joint autoencoder classifier model trained end-to-end. The optimized models have reached an accuracy score of 93.61%. With further investigation on feature contributions to the score, and removing negative effects, we have reached upto 95.94% model accuracy. We also investigated for NOx and SOx emission estimations for different faults and ship loads. Our findings suggested that if we knew the ship load, working conditions and engine health state we could have upto 50% better esti-

mations with full dataset. We also validated our assumption that M0000 produces less emissions. We believe that the joint autoencoder-classifier model will be useful for other time series estimation task on other domains, especially where the operating condition plays a role in the process. The MC90-V dataset has much more initial conditions than we have used in this study. We will be inspecting other scenarios in the future studies. We also plan developing a remaining useful life estimator, which will predict when the failure will occur in the diesel engine. Also we will analyze other types of gas emissions from the engine.

ACKNOWLEDGMENT

This study is funded by the ITU BAP Unit for the MGA-2018-41459 numbered ITU BAP General Research Project (GAP) with the title "Condition Monitoring of Marine Machinery by Machine Learning Methods." The authors also thank to HAVELSAN Naval Combat Management Technologies Center for their support in conducting this research.

REFERENCES

- Back, T., Fogel, D. B., & Michalewicz, Z. (2000). *Evolutionary computation 1: Basic algorithms and operators* (1st ed.). IOP Publishing Ltd.
- Chen, T., & Guestrin, C. (2016). Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*. doi: <https://dx.doi.org/10.1145/2939672.2939785>
- Chollet, F., et al. (2015). *Keras*. <https://keras.io>.
- Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., & Gagné, C. (2012, jul). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research, 13*, 2171–2175.
- Heo, S., & Lee, J. H. (2018). Fault detection and classification using artificial neural networks. *IFAC-PapersOnLine, 51*(18), 470-475. (10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018) doi: <https://doi.org/10.1016/j.ifacol.2018.09.380>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

BIOGRAPHIES



Kürşat İnce received his BSc and MSc degrees from Bilkent University Computer Engineering Department in 1996 and in 1999, respectively. As a software developer, he joined HAVELSAN Inc. in 1996. Currently, he is employed as R&D coordinator in HAVELSAN's Istanbul R&D Center.

Table 5. NOx estimations for different faults and ship loads

\Ship Load	ALL Conditions		FAL		FAU	
Data in use\	MAE	RMSE	MAE	RMSE	MAE	RMSE
FULL Data	0.0213	0.0478	0.0094	0.0337	0.0092	0.0276
M0000	0.0021	0.0030	0.0017	0.0023	0.0018	0.0028
M2503	0.0079	0.0115	0.0066	0.0093	0.0056	0.0076
M2508	0.0164	0.0554	0.0151	0.0575	0.0208	0.0562

Table 6. SOx estimations for different faults and ship loads

\Ship Load	ALL Conditions		FAL		FAU	
Data in use\	MAE	RMSE	MAE	RMSE	MAE	RMSE
FULL Data	0.0053	0.0161	0.0031	0.0084	0.0027	0.0111
M0000	0.0005	0.0009	0.0004	0.0006	0.0005	0.0007
M2503	0.0068	0.0098	0.0053	0.0075	0.0028	0.0038
M2508	0.0039	0.0188	0.0030	0.0114	0.0041	0.0201

Table 7. Mean and standard deviations of NOx emissions for different faults and ship loads

\Ship Load	ALL Conditions		FAL		FAU	
Data in use\	MEAN	STD	MEAN	STD	MEAN	STD
FULL Data	14.1854	1.0699	15.0664	0.5027	13.2622	0.6433
M0000	13.9247	1.0893	15.0275	0.0207	12.8496	0.0390
M2503	14.6247	1.0358	15.4461	0.5228	13.7989	0.7186
M2508	13.9818	0.9141	14.7040	0.4593	13.1176	0.4605

Table 8. Mean and standard deviations of SOx emissions for different faults and ship loads

\Ship Load	ALL Conditions		FAL		FAU	
Data in use\	MEAN	STD	MEAN	STD	MEAN	STD
FULL Data	13.0589	0.4050	13.1646	0.4764	12.9478	0.2718
M0000	12.8619	0.0584	12.9208	0.0041	12.8043	0.0039
M2503	13.4252	0.5029	13.6387	0.5666	13.2101	0.3037
M2508	12.8650	0.1027	12.9179	0.0637	12.8013	0.1046

Mr. İnce started PhD in Gebze Technical University Computer Engineering Department in 2015. Still working on his doctoral thesis, his research interests include machine learning, especially applications of deep learning methods in Industry 4.0. He is also one of the coordinators in Data Istanbul Meetup Group, a community for democratizing machine learning, since 2016.



Gazi Koçak was born in Ankara, Turkey. Graduated from İTÜ Marine Engineering Department by 2003. After working on merchant ships about 2 years he started to work at İTU Maritime Faculty as research assistant. He got scholarship for masters and doctoral studies at Kobe University, Japan. He graduated from doctoral course by 2013. He

is still working at İTUMF as an Assistant Professor.



Yakup Genc received his PhD in Computer Science from the University of Illinois at Urbana-Champaign. Right after graduation, Dr. Genc joined Siemens Corporate Research (SCR) in September 1999. As research scientist, project manager, program

manager and group manager, he developed technology and research strategy in the areas of computer vision, augmented reality and machine learning. His tenure at SCR produced numerous publications and patents. Since September 2012, as a member of the faculty of the Computer Engineering department at the Gebze Technical University, he continues to conduct research in fields of computer vision, augmented reality, autonomous vehicles, machine learning and deep learning while maintaining close ties with the industry for practical applications of his research.

APPENDIX

The sensors that we used in the study is given in Table 9.

Table 9. Sensor list

Feature	Description	Unit/ Range	Feature	Description	Unit/ Range
E02005	ME shaft power (to propeller)	MW	P02072	ME cyl I injection max press (pinjm)	bar
E02006	ME PTO power (to shaftgenerator)	kW	Q02004	ME shaft torque	kNm
E02056	ME cyl I indicated power (IKW)	kW	T00002	FO temp inlet ME	degC
E03760	Shaft power	MW	T01010	HTFW temp inlet ME	degC
G00027	FO flow inlet ME (net flow)	ton/h	T01011	HTFW temp outlet ME	degC
G00108	FO meter volume flow - FO supply	m3/h	T01350	ME LO temp inlet ME	degC
G02011	ME fuel oil consumption	ton/h	T01351	Main LO temp outlet ME	degC
N02015	ME Speed	rpm	T01601	ME air receiver temp	degC
P00023	FO pressure at ME	bar	T01603	ME exh receiver temp	degC
P01005	HTFW press inlet ME	bar	T02014	ME mean cylinder exhaust temp	degC
P01302	Main LO press inlet ME	bar	T02040	ME cyl I exh outlet temp	degC
P01303	Main LO press inlet ME bearings	bar	T02041	ME cyl I exh outlet temp deviation	degC
P01600	ME air receiver press	bar	T02042	ME cyl I air inlet temp	degC
P01602	ME exh receiver press	bar	T02044	ME cyl I oil outlet temp (piston)	degC
P02055	ME Cyl I mean effective pressure (mip)	bar	T04600	TG inlet steam temp (supply line)	degC
P02065	ME cyl I combustion press (pmax)	bar	Z00518	ME exh SOx content	g/kWh
P02066	ME cyl I compression press (pcompr)	bar	Z01970	ME exh NOx content final	g/kWh
P02071	ME cyl I injection open press (pinjo)	bar	Z02013	ME exhaust gas smoke content	%

Physics Informed Neural Network for Health Monitoring of an Air Preheater

Vishal Jadhav¹, Anirudh Deodhar², Ashit Gupta³, and Venkataramana Runkana⁴

^{1,2,3,4}*TCS Research, Tata Consultancy Services Limited, Pune, India*

vi.suja@tcs.com
anirudh.deodhar@tcs.com
ashit.gupta@tcs.com
venkat.runkana@tcs.com

ABSTRACT

Air Preheater (APH) is a regenerative heat exchanger employed in thermal power plants to save fuel by improving their thermal efficiency. Monitoring the health of APH vis-a-vis its fouling is critical because fouling often results in forced outages of the power plant, incurring huge revenue losses. APH fouling is a complex thermo-chemical phenomenon governed by flue gas composition, operating temperatures, fuel type and ambient conditions. Absence of sensors within the APH make it difficult to estimate the level of fouling and its progression even for an experienced operator. Attempts to estimate APH fouling in real-time via modeling are scarce. Here we present a physics-informed neural network (PINN) that tracks the health of an APH by real-time estimation of fouling conditions within the APH as a function of real-time sensor measurements. To account for multi-fluid operation in a multi-sector design of APH, the domain is decomposed into several sub-domains. PINN is applied to each sub-domain and the overall solution is ensured by applying continuity conditions at the sub-domain interfaces. The model predicts the interior temperatures and fouling zones within the APH using external sensor measurements such as air temperature and gas composition. The model predictions are consistent with physics and yet computationally efficient in run-time. The model does not need sensor data but can be improved further by accommodating available sensor data. The real-time predictions by the model improve operator's visibility in fouling. The predictions can be used further for estimating the remaining useful cycle life of the APH, thereby avoiding forced outages. The model can easily be integrated with the digital twin of an APH for its predictive maintenance.

1. INTRODUCTION

Air preheaters (APH) are used in thermal power plants for improving thermal efficiency by recovering the excess heat from boiler exhaust gases. APH fouling is a serious and recurring problem that often causes unplanned outages of the plant incurring huge revenue losses. Complex thermo-chemical phenomena in fouling and lack of sensors within APH, make it difficult to monitor the fouling in real-time requiring predictive models.

APH typically comprises two or three successive layers of matrix that enable effective heat transfer by increasing the surface area per unit volume. This rotating metallic matrix extracts heat from the hot flue gas and passes it on to the ambient air flowing in a countercurrent manner with respect to the gas. Depending upon the number of air streams, APH can have a 2-sector or a 3-sector arrangement. Fouling is caused by gradual deposition of a chemical compound called ammonium bisulfate (ABS), formation of which is predominantly influenced by the internal temperature profile within APH and the gas composition (ammonia NH₃, sulfur trioxide SO₃ and ash). Several ABS formation and deposition studies (Muzio, Bogseth, Himes, Chien, & Dunn-Rankin, 2017; Menasha, Dunn-Rankin, Muzio, & Stallings, 2011; Zhou, Zhang, Deng, & Ma, 2016) have revealed that the gas temperature profile within APH influences not only the magnitude of fouling but also the location of fouling and in turn governs its overall progression. Although no models have been developed for estimating chemical formation and deposition directly, several models capturing thermal phenomena have been developed based on first principles including Computational Fluid Dynamics (CFD) (Li, 1983; Skiepkio, 1988; Drobnic, Oman & Tuma 2006, Wang, Bu, Li, Tang, & Che, 2019; Heidari-Kaydan, Hajidavalloo, & Mehrzad, 2021). However, most of the models are computationally expensive with significantly high inference time and hence not amenable for real-time applications.

Vishal Jadhav et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Empirical models proposed for estimating propensity of fouling (Burke & Johnson 1982; Wang et al. 2019; Chen, Xu, Yang, Wang, & Wang 2020) also require the internal temperature profile of APH for accurate estimations. Data-driven models based on machine or deep learning have been explored for fouling estimation (Sundar, Rajagopal, Zhao, Kuntumalla, Meng, Chang, Shao, Ferreira, Miljkovic, Sinha, & Salapaka, 2020; and Gupta, Jadhav, Patil, Deodhar, & Runkana, 2021). However, these models are heavily dependent on the quality and availability of data. In absence of sensor measurements inside the APH and overall difficulty in obtaining sufficient industrial data, purely data-driven models are not effective. In addition, often the inferences from purely data-driven models may defy physical principles.

Physics informed neural networks (PINNs) (Raissi, Perdikaris, & Karniadakis 2019) are gaining popularity as a viable alternative to address limitations and harness the power of both physics-based and data-driven models. PINNs are a type of universal function approximators that are trained by imposing governing partial differential equations as constraints. These constraints are applied by introducing governing equation residuals and boundary or initial conditions in the loss function. This approach enables the neural network to incorporate domain knowledge in the learning process and learns virtually without any data in a semi-supervised manner. PINNs make the model more flexible by eliminating a fixed mesh typically required in a physics-based numerical solver, while remaining physically consistent.

Lagaris, Likas and Fotiadis (1998) were the first to introduce neural networks to solve boundary value problems for partial differential equations (PDEs). Raissi, et al. (2019) introduced the concept of incorporating governing PDEs in the loss function of a deep neural network to solve both forward and inverse problems. Since then, PINN has been used for solving various scientific problems in several domains (Cuomo, Di Cola, Giampaolo, Rozza, Raissi, & Piccialli, 2022) including fluid flow (Cai, Mao, Wang, Yin & Karniadakis, 2022) and heat transfer (Cai, Wang, Wang, Perdikaris & Karniadakis, 2021).

A few limitations of the PINNs have been recently highlighted by Cuomo et al. (2022) in their review. Even though the inference time for PINN models is considerably low, high training time and significant convergence difficulties in complex scenarios limit their implementation in real life applications (Jagtap & Karniadakis, 2020). Shukla, Jagtap and Karniadakis (2021) suggested the distributed framework for training PINN models to reduce the training time. Domain decomposition is one such strategy of distributed framework usually adapted to reduce the complexity of training PINNs (Heinlein, Klawonn, Lanser, & Weber, 2021). cPINN (Jagtap, Kharazmi, & Karniadakis, 2020) and xPINN (Jagtap & Karniadakis, 2020) networks employ the domain decomposition strategy to get accurate solutions of complex nonlinear conservation laws. Recently,

Moseley, Markham, and Nissen-Meyer (2021) proposed the domain decomposition approach to solve large multiscale problems. Another limitation of current PINN techniques is that they fail to generalize over dynamically changing boundary conditions (Cuomo et al. 2022) for governing differential equations, a scenario often found in industrial applications.

PINNs trained over single set of boundary conditions cannot be used in application where parameter values change dynamically (Wang, Planas, Chandramowlishwaran and Bostanabad, 2021). Wang et al. (2021) proposed a ‘train once use forever’ algorithm comprising of a combination of GFNet and Mosaic Flow Predictor that enables one time training of a neural network that can generalize over arbitrary boundary conditions as well as arbitrary domain shapes. Meta learning (Penwarden, Zhe, Narayan, & Kirby, 2021) and hypernetwork (Belbute-Peres, Yi-fan, & Fei, 2021) approaches have also been suggested for adapting PINNs to dynamic boundary conditions. Chakraborty (2021) suggested the use of transfer learning for training of multi fidelity PINNs. Desai, Mattheakis, Joy, Protopapas, and Roberts, (2021) has presented use of transfer learning with pre-trained neural network for one-shot inference for linear system of both ordinary and partial differential equations.

In the present work, we apply some of these concepts to develop a PINN model for real-time health monitoring of an industrial APH. The base PINN model is developed by decomposing the APH into three sub-domains and stitching the individual sub-domain PINNs by applying continuity conditions at the respective interfaces. The model solves a set of two-dimensional governing equations for capturing the heat transfer phenomenon and predicts the internal temperature profile of APH for air, gas and metal based on the external boundary conditions such as inlet air and gas temperatures. Boundary conditions used to solve governing equations herein refers to inlet gas and air temperatures, which are typically known through sensor measurements. Fouling propensity is a function of temperatures and chemical concentrations in APH. Online monitoring of fouling propensity can be enabled through PINN models trained for different boundary conditions. However, in online applications temperatures of gas and air, flow rates and composition of flue gas vary significantly. This results in numerous combinations of conditions and for each such condition offline simulation or training of PINN is not practical. To address this challenge, a transfer learning framework is used which enables computationally inexpensive and near real-time re-training and inference from the network for a change in boundary conditions, making the model suitable for real-time industrial application. The model inference is shown to be as accurate as and significantly faster than corresponding physics-based numerical solution. In current study finite combination of temperatures (9 cases) are considered to demonstrate the use of transfer learning to speed up the training time. However, in practice numerous

combinations of temperatures or other parameters are possible. Transfer learning framework proposed will be useful for online estimation of temperature profile and fouling propensity monitoring in industrial application. Online estimation of temperature profile further can also be used for digital twin applications wherein real time predictions are used for process optimization, maintenance decisions, safety related decisions along with monitoring.

2. METHODOLOGY

2.1. Governing Equations for APH heat transfer

Figure 1 shows the schematic of APH of height H , outer diameter d_o , inner diameter d_i and sector angles of $\beta_g, \beta_{a1}, \beta_{a2}$ for flue gas, primary air, and secondary air flow. Flue gas enters APH from the top whereas primary air and secondary air enter from bottom. The metallic matrix rotates at ω revolutions per minute (rpm).

High temperature flue gas enters from top and heats the matrix, which in turn rotates and transfers this heat to the cold ambient air entering from bottom. While convection dominates the heat transfer between fluids and metal, conduction contributes significantly to the heat transfer within metal. In the current work, we consider a two-dimensional formulation (tangential and axial direction) for solving thermal governing equations inside the APH. Heat transfer in the radial direction is assumed to be constant and hence not accounted for in the governing equations (Skiapko 1988).

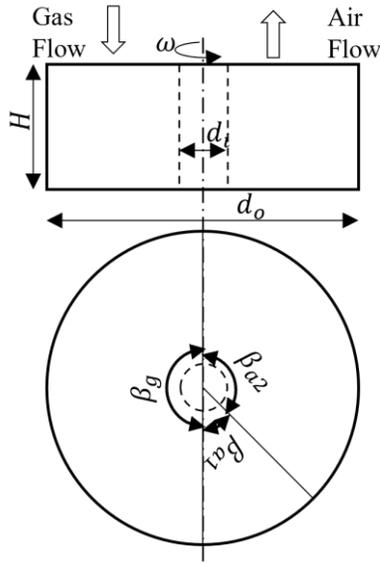


Figure 1 APH Schematic

The computational domain considered is shown in Figure 2. Due to the presence of different fluid (gas & air) channels in

the tangential direction, fluid temperature profile would be discontinuous. PINN models are known to encounter difficulties in generalizing for such solution discontinuities (Jagtap et al. 2020). Therefore, entire domain (Ω) is divided into three subdomains presented as gas side subdomain (Ω_g), primary air side sub domain (Ω_{a1}), and secondary air side sub domain (Ω_{a2}), as shown in Figure 2. Each sub domain coordinates are normalized from 0 to 1 for both axial and tangential directions. It can be noted that, for gas side subdomain positive axial direction is from top to bottom which is same as gas flow direction. Similarly, for primary air side and secondary air side subdomain, positive axial direction is from bottom to top which is same as flow direction for primary and secondary air. Positive tangential direction is considered from left to right for all subdomains, which is same as rotational direction of matrix. For simplicity, all metal matrix layers are assumed to be made of a single homogeneous material, however the approach mentioned can be extended to a multi material matrix APH as well.

Eqs. (1-6) represent the non-dimensionalised governing equations for heat transfer between the fluids and the metal (Skiapko 1988). Here, subscript $m, g, a1, a2$ are used to represent matrix, gas, primary air and secondary air respectively. Subscript m_g, m_{a1}, m_{a2} are used to represent matrix in gas, primary air and secondary air domain respectively. Number of transfer units (NTU) and Peclet number (Pe), are used to non dimensionalize the governing equations (Skiapko 1988).

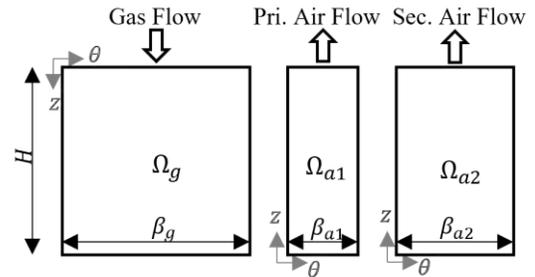


Figure 2 Computational domain for Air Preheater

$$\frac{\partial T_{m_g}}{\partial \theta} = NTU_{m_g} (T_g - T_{m_g}) + Pe_{m_g}^{-1} \frac{\partial^2 T_{m_g}}{\partial z^2} \quad (1)$$

$$\frac{\partial T_g}{\partial z} = NTU_g (T_{m_g} - T_g) \quad (2)$$

$$\frac{\partial T_{m_{a1}}}{\partial \theta} = NTU_{m_{a1}} (T_{a1} - T_{m_{a1}}) + Pe_{m_{a1}}^{-1} \frac{\partial^2 T_{m_{a1}}}{\partial z^2} \quad (3)$$

$$\frac{\partial T_{a1}}{\partial z} = NTU_{a1}(T_{m_{a1}} - T_{a1}) \quad (4)$$

$$\frac{\partial T_{m_{a2}}}{\partial \theta} = NTU_{m_{a2}}(T_{a2} - T_{m_{a2}}) + Pe_{m_{a2}}^{-1} \frac{\partial^2 T_{m_{a2}}}{\partial z^2} \quad (5)$$

$$\frac{\partial T_{a2}}{\partial z} = NTU_{a2}(T_{m_{a2}} - T_{a2}) \quad (6)$$

Temperature measurements of gas and air at the respective inlets are used as boundary conditions to solve the governing equations. Eqs. (7-9) represent the boundary conditions.

$$T_g(\theta, z = 0) = T_{gin} \quad (7)$$

$$T_{a1}(\theta, z = 0) = T_{a1in} \quad (8)$$

$$T_{a2}(\theta, z = 0) = T_{a2in} \quad (9)$$

Along with boundary conditions, continuity constraints due to rotation of matrix from gas side to primary air side, primary air side to secondary air side and secondary air side to gas side again are applied as shown in eqs. (10-12)

$$T_{m_g}(\theta = 0, z) = T_{m_{a2}}(\theta = 1, 1 - z) \quad (10)$$

$$T_{m_g}(\theta = 1, z) = T_{m_{a1}}(\theta = 0, 1 - z) \quad (11)$$

$$T_{m_{a1}}(\theta = 1, z) = T_{m_{a2}}(\theta = 0, z) \quad (12)$$

As axial heat conduction in metal is zero at the end of metallic layer additional matrix temperature gradient constraints are imposed as shown eqs. (13-15)

$$\frac{\partial T_{m_g}[\theta, (z = 0 \text{ and } 1)]}{\partial z} = 0 \quad (13)$$

$$\frac{\partial T_{m_{a1}}[\theta, (z = 0 \text{ and } 1)]}{\partial z} = 0 \quad (14)$$

$$\frac{\partial T_{m_{a2}}[\theta, (z = 0 \text{ and } 1)]}{\partial z} = 0 \quad (15)$$

2.2. Physics Informed Neural Network

2.2.1. Neural Network Architecture

Base PINN model for APH consists of a deep neural network (DNN) for each of the subdomains p in APH (i.e., $\Omega_g, \Omega_{a1}, \Omega_{a2}$). The spatial co-ordinates (θ, z) are the inputs to each network and the outputs are fluid temperature (T_f) (air or gas) and matrix temperature (T_m) for each sub-domain respectively. Let $\mathcal{N}^L: \mathbb{R}^{D_i} \rightarrow \mathbb{R}^{D_o}$ be a deep neural network of L layers and N_k neurons in k^{th} layer ($N_0 = D_i$ and $N_L = D_o$). The weight matrix and bias vector in the k^{th} layer ($1 \leq k \leq L$) are denoted by $\mathbf{W}^k \in \mathbb{R}^{N_k \times N_{k-1}}$ and $\mathbf{b}^k \in \mathbb{R}^{N_k}$ respectively. Input vector is denoted as $\mathbf{x} \in \mathbb{R}^{D_i}$, output vector at k^{th} layer is denoted as $\mathcal{N}^k(\mathbf{x})$ and $\mathcal{N}^0(\mathbf{x}) = \mathbf{x}$. Activation function is denoted as Φ . DNN is defined by eq. (16):

$$\mathcal{N}^k(\mathbf{x}) = \Phi(\mathbf{W}^k \mathcal{N}^{k-1}(\mathbf{x}) + \mathbf{b}^k), \quad 1 \leq k \leq L \quad (16)$$

Let, $\Theta = \{\mathbf{W}^k, \mathbf{b}^k\}$ be a collection of all weights and biases. Then output of neural network is given by eq. (17).

$$u_{\Theta}(\mathbf{x}) = \mathcal{N}^L(\mathbf{x}; \Theta) \quad (17)$$

Figure 3 shows the schematic of PINN architecture, wherein three deep neural networks are used for each sub-domain respectively. For each sub domain, the output of individual deep neural network is given as

$$u_{i_{\Theta}}(\mathbf{x}) = \mathcal{N}_i^L(\mathbf{x}; \Theta), \forall i = 1, 2, 3 \quad (18)$$

The final solution will be given as

$$u_{\Theta}(\mathbf{x}) = \bigcup_{i=1}^3 u_{i_{\Theta}}(\mathbf{x}) \quad (19)$$

2.2.2. Sub domain Loss Function

Total loss for PINN comprises of mean squared error (MSE) due to residuals of governing equations calculated using collocation points ($MSE_{\mathcal{F}_p}$), loss due to boundary condition calculated using boundary points (MSE_{bc_p}), MSE loss due to interface condition calculated at interface points (MSE_{ic_p}) and MSE loss due to matrix temperature gradient calculated at top and bottom points of each sub domain (MSE_{grad_p}). Mean squared error for different components of sub domain p is calculated using eqs. (20-23):

$$MSE_{\mathcal{F}_p} = \frac{1}{N_{\mathcal{F}_p}} \sum_{i=1}^{N_{\mathcal{F}_p}} \left| \mathcal{F}(\theta_{\mathcal{F}_p}^i, z_{\mathcal{F}_p}^i) \right|^2 \quad (20)$$

$$MSE_{ic_p} = \frac{1}{N_{ic_p}} \sum_{i=1}^{N_{ic_p}} \left| T_{m_p}(\theta_{ic_p}^i, z_{ic_p}^i) - T_{m_{p^+}}(\theta_{ic_p}^i, z_{ic_p}^i) \right|^2 \quad (21)$$

$$MSE_{bc_p} = \frac{1}{N_{bc_p}} \sum_{i=1}^{N_{bc_p}} \left| T_{f_p}^i - T_{f_p}(\theta_{bc_p}^i, z_{bc_p}^i) \right|^2 \quad (22)$$

$$MSE_{grad_p} = \frac{1}{N_{grad_p}} \sum_{i=1}^{N_{grad_p}} \left| \frac{\partial T_{m_p}(\theta_{grad_p}^i, z_{grad_p}^i)}{\partial z} \right|^2 \quad (23)$$

Where, \mathcal{F} is the residual of governing PDEs, subscript p^+ indicates the neighboring subdomain to subdomain p , $N_{\mathcal{F}_p}, N_{ic_p}, N_{bc_p}, N_{grad_p}$ represents number of collocation points, number of interface condition points, number of boundary condition points and number of matrix temperature gradient condition points in p^{th} subdomain respectively. $(\theta_{\mathcal{F}_p}^i, z_{\mathcal{F}_p}^i)$, $(\theta_{bc_p}^i, z_{bc_p}^i)$ and $(\theta_{grad_p}^i, z_{grad_p}^i)$ represents the co-ordinates of the residual points, boundary condition points and gradient condition points for p^{th} sub domain. $(\theta_{ic_p}^i, z_{ic_p}^i)$ represents the common interface points of two neighboring subdomains p and p^+ . Loss for p^{th} subdomain is given in eq. (24).

$$\mathcal{L}(\Theta)_p = MSE_{\mathcal{F}_p} + MSE_{ic_p} + MSE_{bc_p} + MSE_{grad_p} \quad (24)$$

Total loss for PINN is given by eq. (25), where subscript $g, a1, a2$ represents subdomain for gas, primary air, and secondary air.

$$\mathcal{L}(\Theta) = \mathcal{L}(\Theta)_g + \mathcal{L}(\Theta)_{a1} + \mathcal{L}(\Theta)_{a2} \quad (25)$$

DNN for each subdomain consist of one input layer (two neurons), two hidden layers (with 16 neurons in each layer) and one output layer (two neurons). Activation function used for hidden layers and output layer is tanh. Gradients for evaluating the residual equation are calculated using auto differentiation feature (Baydin, Pearlmutter, Radul, & Siskind, 2018). Adam optimizer is used to train PINN model mean squared error is used as loss metric. Additionally, reducing learning rate callback and early stopping callback features from tensorflow were used for better control while training. If the loss does not reduce compared to the best loss value for 50 epochs, learning rate is reduced by factor of 0.1 with reducing learning rate callback. Also, early stopping callback was used to stop the training if the training loss does not improve for 100 epochs. Machine used for numerical simulation and PINN model training has specifications as follows: AMD Ryzen 5 2500U processor with Radeon Vega Mobile Gfx 2.00 GHz, RAM of 24 GB and 64-bit operating system.

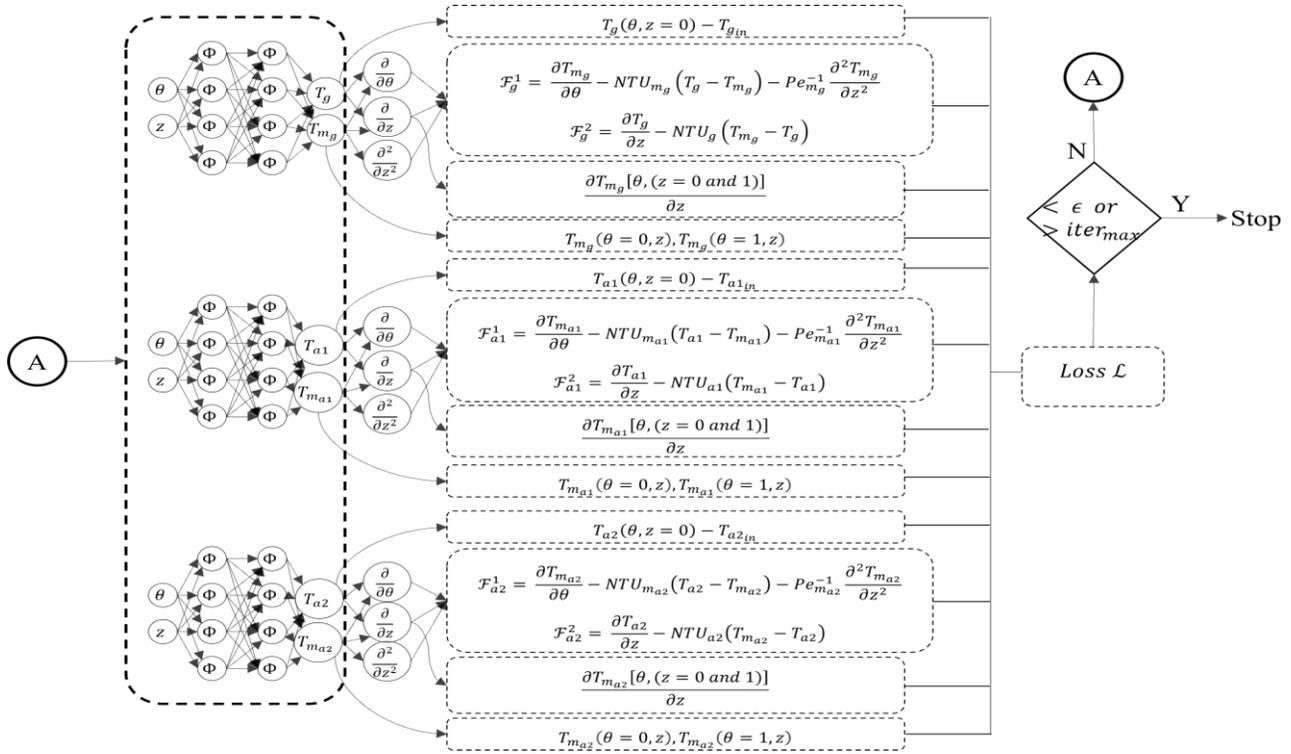


Figure 3 PINN Architecture

2.3. Transfer Learning Based Dynamic Prediction using PINN

The base PINN described above can predict the internal temperatures accurately only for the set of boundary conditions it is trained with. However, industrial scenario requires a model that can predict the temperatures dynamically for varying boundary conditions in near real-time. To enable quick re-training and inference for a different boundary condition, a transfer learning inspired approach is utilized. The weights of the base PINN model trained for benchmark boundary condition are used for initialization and instead of training all those weights, weights and bias for the input layer and first hidden layer are frozen for each DNN of each subdomain. This follows from the assumption that layer 1 of each DNN captures the benchmark physical phenomena sufficiently and the second layer manipulates the inference for the altered boundary condition. The accuracy and computational time based on this method is later compared with the traditional physics-based numerical solver and the PINN model with all layers trainable.

2.4. Health Monitoring using inference from PINN model

The internal temperature profile of APH plays a crucial role in determining the progression of fouling as it influences the chemical reactions and more importantly the location of ABS deposition zone within the APH. However, in absence of any sensors inside APH the operators remain blind to the fouling phenomena unfolding inside the matrix. Estimation of fouling propensity and identification of ABS deposition zone can assist the operator to make informed decisions about managing the fouling vis-a-vis the operating and maintenance costs it incurs.

Chen et al. (2020) have suggested a number that indicates the propensity of fouling within APH based on the temperatures and the gas composition (ammonia NH_3 and sulfur oxide SO_3). PINN developed above can be utilized not only for estimation of fouling propensity but also for identifying fouling deposition zone within APH. Localization of this fouling zone is critical because it moves with the internal temperature conditions and has a large influence on the overall health of APH. Typically, when this fouling zone is close to the gas exit, the deposits within it are removable by a cleaning equipment called soot blower. However, when this fouling zone moves away from gas exit due to a shift in temperatures, it increases the risk of APH clogging and ultimate forced outage of the plant. ABS deposition temperature depends on the concentration of NH_3 and SO_3 in the APH. It can be calculated using the empirical relation presented in eq. (26) (Huang, Sun, Chen, Li, Gu, Hu, & Cheng 2015).

$$T_{ABS} = 0.4059[\ln(\varphi_{NH_3}\varphi_{SO_3})]^2 + 11.45 \ln(\varphi_{NH_3}\varphi_{SO_3}) + 192.29 \quad (26)$$

Here, T_{ABS} represents initial condensation temperature of ABS in $^{\circ}C$; φ_{NH_3} and φ_{SO_3} represents the concentration of NH_3 and SO_3 in ppm.

Fouling propensity indicator (R-Number) for ABS deposition tendency is given by eq. (27) (Chen et al. 2020). Here, $\varphi_{NH_3,0}$ and $\varphi_{SO_3,0}$ represents the reference concentrations of NH_3 and SO_3 in the flue gas at the inlet of the air preheater, respectively taken as 3 ppm and 5 ppm; T_{ABS} in K, $T_{cold,bottom,avg}$ and $T_{cold,top,avg}$ are the average temperature at the cold end of the APH in K, similarly $T_{hot,bottom,min}$ and $T_{hot,top,min}$ are the average temperature at the hot end of the APH in K.

$$R - number = \frac{\varphi_{NH_3}\varphi_{SO_3}}{\varphi_{NH_3,0}\varphi_{SO_3,0}} \frac{T_{ABS}-T_{cold,bottom,avg}}{T_{cold,top,avg}-T_{cold,bottom,avg}} \exp\left(\frac{T_{ABS}-T_{hot,bottom,min}}{T_{hot,top,min}-T_{hot,bottom,min}}\right) \quad (27)$$

Figure 4 represents the summary of steps required to perform online health monitoring of the APH. A prerequisite base PINN model can be trained offline for single set of boundary conditions and given design parameters (geometry, material properties) of APH. This model is then further used in online health monitoring by freezing the weights between input and first layer of the networks. In online conditions, boundary conditions change continuously and hence the internal temperature profile in APH change as well. To infer the internal temperature profile, transfer learning framework is used along with base PINN model to train the new PINN model corresponding to new boundary conditions. PINN model networks are used to predict the internal temperature profile in APH. Internal temperature profile is then used to estimate unremovable deposit region and fouling propensity. Insight of deposit region and fouling propensity can be used by operator to do suitable operation changes.

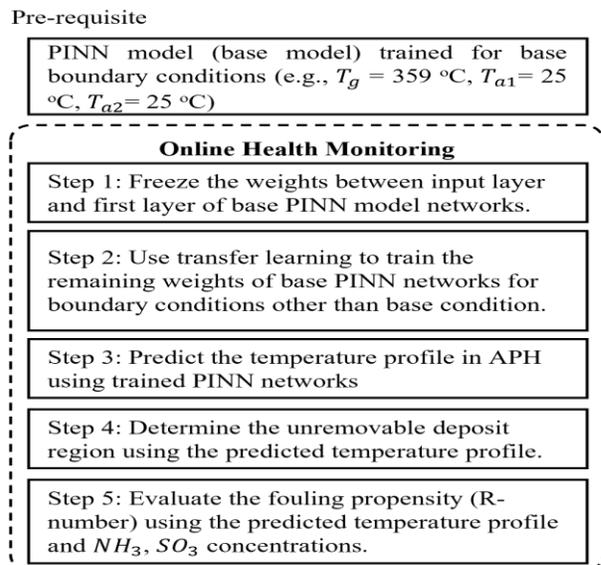


Figure 4 Flow chart for health monitoring using PINN

3. RESULTS AND DISCUSSION

3.1. Base PINN and Numerical Solution comparison

Numerical solution of the governing equations for the APH is obtained using finite difference method described by Li (1983). The numerical solution was validated with experimental data earlier (Gupta et al. 2021). Design and material related parameter values of APH used for the simulation are mentioned in Appendix A.1. This numerical simulation is used as a benchmark for comparing predictions from PINN models. The base PINN model is trained for the same APH with inlet gas temperature as 359°C and inlet air temperature as 25°C. It is trained using the architecture mentioned in section 2.2. It uses 15000 collocation, 15000 boundary and 15000 sub-domain interface points for training. As shown in Figure 5 the base PINN is trained for 788 epochs to get an acceptable loss value. Training was stopped as per early stopping criteria (no improvement in total loss for last 100 epochs). Model weights were restored to best weights obtained corresponding to best loss value. Post training of the model, values of temperature for flue gas side, primary air side, secondary air side and the matrix are inferred. Comparison of temperature profiles obtained through numerical finite difference method and PINN model is presented in Figure 6. The top row shows the fluid temperature profiles; the middle row shows the metal matrix temperature profiles. For continuity, the three sub-domains gas, primary air and secondary air are connected at the interfaces in Figure 6. The region between $\theta = 0-180^\circ$ represents gas domain, $\theta=180^\circ-250^\circ$ represents primary air and $\theta= 250^\circ-360^\circ$ represents secondary air. As seen in the figure, there is an excellent match between the two solutions. The base PINN solution has a mean absolute error of $8.1e-3$ and a maximum absolute error of 0.03 when compared against the numerical solution for normalized value. It is equivalent to mean absolute error of 2.7°C and maximum absolute error of 10°C considering non-normalized solution. The domain decomposition technique also enabled capturing the thermal phenomena near the interfaces without any loss of accuracy. Although PINN requires a considerable time for training, its inference time is significantly lower than the numerical solution as shown in Table 1. The inference time of a trained PINN model will not deviate significantly even if the required granularity (mesh size) of the solution is altered. The same cannot be said about the numerical solution. Therefore, PINNs can be effectively used for soft sensing temperatures within APH in a near real-time scenario. The next challenge of making PINN work for varying boundary conditions is addressed in the next section.

Table 1 Training and Inference Time comparison for Numerical Method and PINN solution for Base Case

	Numerical Method	PINN
Training Time (sec)	N/A	4212
Inference Time (sec)	1076	1.8

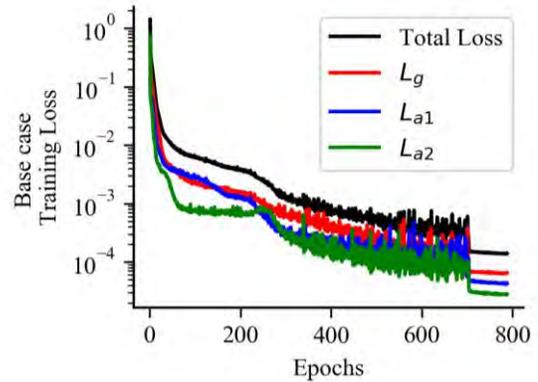


Figure 5 Training Loss for Base Case (Inlet Gas Temperature 359°C, Inlet Air Temperature 25°C)

3.2. Training and inference for dynamic boundary condition

As discussed in section 2.3, the base PINN network with flue gas inlet temperature as 359°C and air inlet temperature as 25°C is used for training new PINNs for different boundary conditions via the transfer learning-based approach. The benefits of the approach are evaluated by testing it with a set of 9 different boundary conditions often encountered in industrial APH operation. The accuracy and the training - inference time for the transfer learning approach is compared with the corresponding numerical simulation as well as a PINN trained from scratch for the given boundary condition, as shown in Table 2. As seen, there is no loss of accuracy with the transfer learning approach when compared to PINN from scratch and the numerical simulation, validating its application. As the requirement is that of monitoring APH health in near real-time, the combined training and inference time of PINNs is compared against the inference time of a numerical solution (as it does not need any retraining). The premise here is that the PINN model will be trained online and used immediately for temperature predictions.

The training and inference time for a PINN trained from scratch typically exceeded the inference time required for a numerical simulation. However, a transfer-learned PINN adapted for a new boundary condition using the previously built base PINN brought down the training and inference time substantially compared to the numerical simulation. On an average the combined training and inference time for a transfer-learned PINN was 78% less than the corresponding inference time for the numerical simulation.

Industrial digital twins are increasingly using edge analytics for reducing the traffic and costs for cloud computations (Sánchez, Jörgensen, Törngren, Inam, Berezovskyi, Feng, Fersman, Ramli, & Tan 2021). Often the task of re-training of machine/deep learning models is allocated to cloud due to

intensive computational requirements and dependence on past historical data (which is stored on cloud). PINNs can help reduce the dependence on large quantities of data and a quick retraining framework such as this one can enable it to run on the edge instead of on cloud. This online retraining and near real-time predictions from the retrained model can provide an effective way of monitoring equipment health in industrial and manufacturing settings.

3.3. Health monitoring of APH using PINNs

Near real-time temperature predictions from PINN model can be effectively used for monitoring the risk posed by fouling. As an example, the internal temperatures predicted by PINN for the base case are used for calculating the fouling propensity and identifying the fouling zone as described in section 2.4. The fouling propensity is calculated assuming ammonia (NH_3) and sulfur oxide (SO_3) concentrations as 3 ppm and 10 ppm respectively. For this scenario, T_{ABS} calculated using eq. (26) is 236.45°C and R-number calculated using eq. (27) is 0.79. Figure 7 shows the maximum, minimum and average gas temperature profile in the axial direction against the depth of APH ($z = 0$ indicates gas entry). The overall fouling zone for the given conditions is also indicated based on the calculated T_{ABS} . As seen in the figure a portion of this fouling zone falls beyond the reach of the cleaning soot blowing equipment and hence creating a

permanent deposit inside the APH. Under normal circumstances the operator is completely blind to this insight. However, with this estimation operator may take some corrective actions to decelerate the fouling.

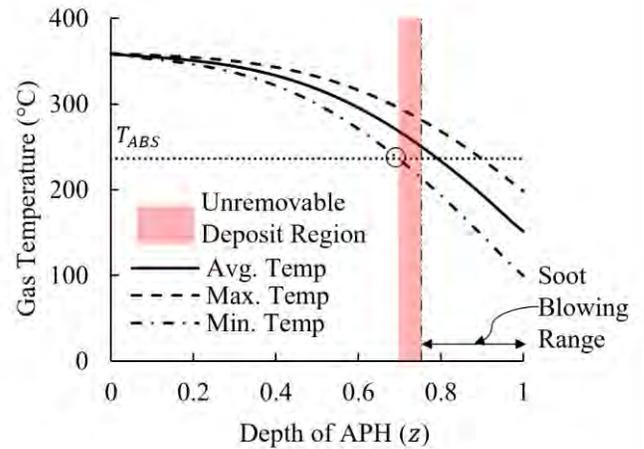


Figure 7 Gas Temperature distribution and Unremovable Deposit region in APH

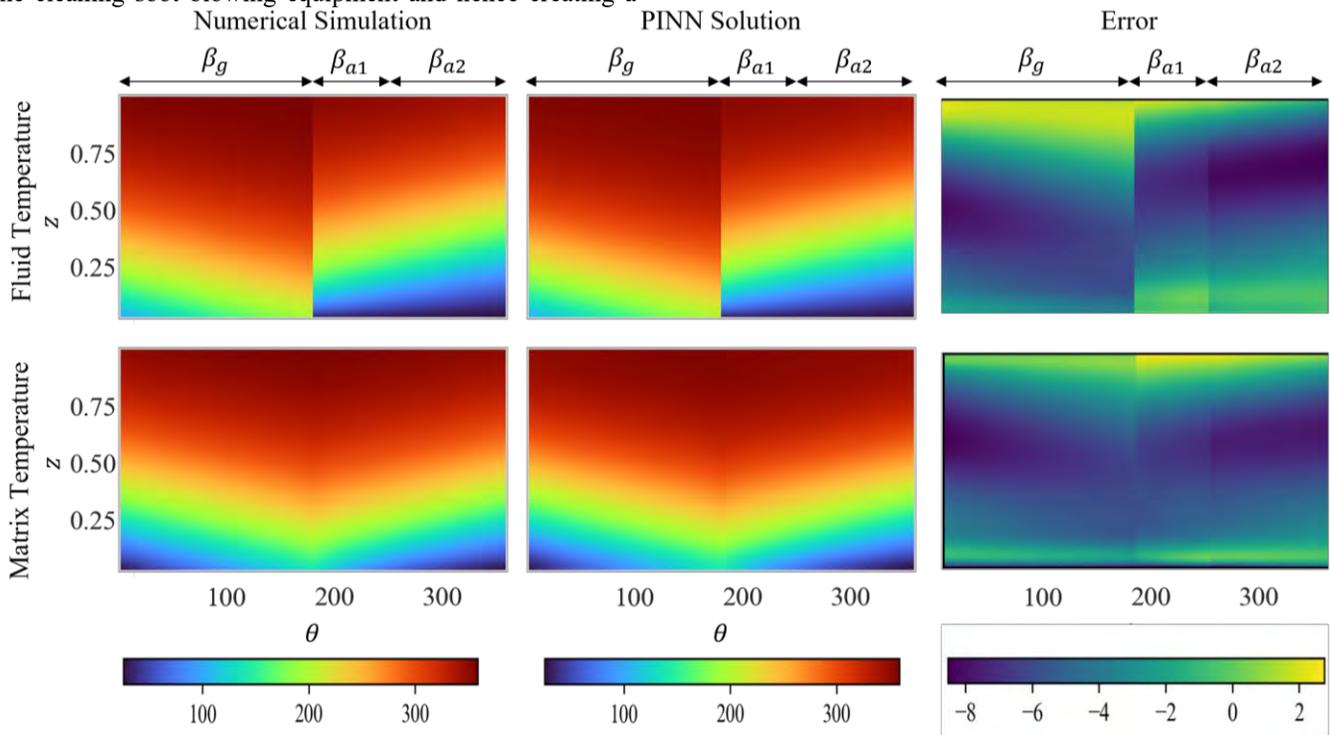


Figure 6 Comparison of PINN and Numerical Solution

Table 2 Comparison of inference time for Transfer Learning based Training of PINN and numerical simulation

Case	Inlet Gas Temperature BC	Inlet Air Temperature BC	Numerical Simulation	PINN trained with random initialization (without using the Base PINN)			Transfer Learning using the Base PINN		
			Inference time (sec)	Train time (sec)	Inference time (sec)	MAE (wrt to corresponding numerical simulation)	Train time (sec)	Inference time (sec)	MAE (wrt to corresponding numerical simulation)
1	329	19	1028	3716	3.6	0.012	211	3.5	0.0105
2	354	48	922	2782	3.7	0.005	255	3.8	0.0052
3	331	38	1122	3314	3.9	0.030	187	3.6	0.0098
4	330	31	915	3174	4.0	0.008	190	3.4	0.0105
5	390	36	1066	2895	3.5	0.007	194	3.5	0.0033
6	320	37	1076	2847	3.3	0.016	268	3.5	0.0112
7	399	23	1179	3370	3.7	0.011	194	3.6	0.0056
8	388	13	1188	2878	3.7	0.020	195	3.9	0.0021
9	363	43	1038	2726	3.7	0.016	181	3.7	0.0043

Figure 8 shows an example of effect of ammonia and ambient inlet air temperature on the fouling risk for APH. It is seen that high ammonia and low ambient temperature condition poses the greatest risk because of the formation of deep unremovable deposits within APH. The PINN model therefore can be used as an effective monitoring tool for APH predictive maintenance.

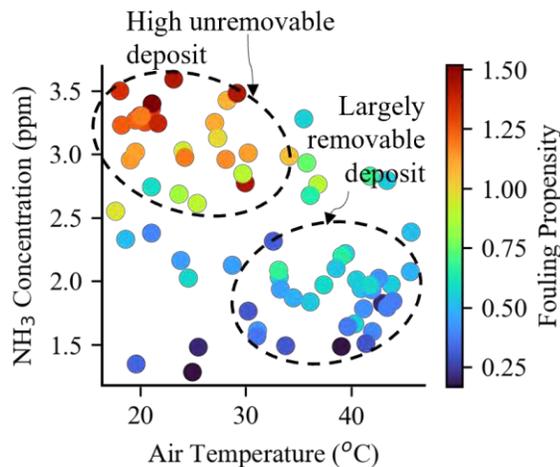


Figure 8 Fouling Propensity Monitoring

Although the PINN framework demonstrated here is certainly very useful, it can be improved and enhanced further to make it more effective. Firstly, the model can be modified to incorporate equations governing the chemical reactions and the deposition kinetics as well. Although the

model demonstrated in this work caters to dynamic inlet temperature conditions, it can be expanded to accommodate dynamic flow rates, material properties as well as APH geometries. This can pave the way for building a generic and adaptable PINN model for monitoring and predictive maintenance of APH. On the deep learning front, we plan to explore different methods for building an all-condition PINN model capable of handling dynamic changes without having to do extensive retraining.

4. CONCLUSION

A Physics-informed Neural Network (PINN) is designed for capturing thermal phenomena in an air preheater (APH) used in thermal power plants. The APH is divided into three sub-domains and a separate deep neural network (DNN) is constructed for each of them. The base PINN model is trained by stitching the three DNNs together through a common loss function comprising of governing partial differential equations and continuity constraints at the sub-domain interfaces. Further, a transfer learning framework is used to enable quick training and inference from the PINN model for dynamically changing boundary conditions. The PINN model is shown to be faster than corresponding physics-based numerical solver, without appreciable loss of accuracy, making the model suitable for online and real-time applications. The predictions from the model are further used for estimating propensity of fouling in APH in near real-time, thereby assisting the operator in avoiding forced outages by taking informed decisions. The proposed PINN framework can be easily integrated into a digital twin of APH for a predictive maintenance application.

ACKNOWLEDGEMENT

The current work was funded by Tata Consultancy Services Limited. The authors thank Mr. K. Ananth Krishnan, and Dr. Gautam Shroff for their encouragement and support. The authors would also like to thank Dr Shirish Karande, Dr Lovekesh Vig and Mr. Souridas A. for their inputs and suggestions.

REFERENCES

- Baydin, A.G., Pearlmutter, B.A., Radul, A.A. & Siskind, J.M., (2018). Automatic Differentiation in Machine Learning: a Survey. *Journal of Machine Learning Research*, 1-43.
- Belbute-Peres, F.D.A., Chen, Y.F. & Sha, F., (2021). HyperPINN: Learning parameterized differential equations with physics-informed hypernetworks. *arXiv preprint arXiv:2111.01008*.
- Burke, J.M. & Johnson, K.L., (1982). Ammonium sulfate and bisulfate formation in air preheaters. *Final report Oct 80-Oct 81 (No. PB-82-237025)*. Radian Corp., Austin, TX (USA).
- Cai, S., Wang, Z., Wang, S., Perdikaris, P. & Karniadakis, G.E., (2021). Physics-Informed Neural Networks for Heat Transfer Problems. *Journal of Heat Transfer*, 143(6). doi:10.1115/1.4050542
- Cai, S., Mao, Z., Wang, Z., Yin, M. & Karniadakis, G.E., (2022). Physics-informed neural networks (PINNs) for fluid mechanics: a review. *Acta Mechanica Sinica*. doi:10.1007/s10409-021-01148-1
- Chakraborty, S. (2021). Transfer learning based multi-fidelity physics informed deep neural network. *Journal of Computational Physics*. doi:https://doi.org/10.1016/j.jcp.2020.109942
- Chen, X., Xu, S., Yang, Y., Wang, L.M. & Wang, D.D., (2020). Performance Analysis and Optimization of Flue Gas Waste Heat Recovery System of a 600MW Coal-Fired Power Plant. *ASME 2020 Power Conference*. doi:10.1115/POWER2020-16873
- Cuomo, S., Di Cola, V.S., Giampaolo, F., Rozza, G., Raissi, M. & Piccialli, F., (2022). Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next. *arXiv preprint arXiv:2201.05624*.
- Desai, S., Mattheakis, M., Joy, H., Protopapas, P. & Roberts, S., (2021). One-Shot Transfer Learning of Physics-Informed Neural Networks. *arXiv preprint arXiv:2110.11286*.
- Drobnic, B., Oman, J., & Tuma, M. (2006). A numerical model for the analyses of heat transfer and leakages in a rotary air preheater. *International Journal of Heat and Mass Transfer*, 5001-5009.
- Gupta, A., Jadhav, V., Patil, M., Deodhar, A. & Runkana, V., (2021). Forecasting of Fouling in Air Pre-Heaters Through Deep Learning. *ASME 2021 Power Conference*. doi:10.1115/POWER2021-64665
- Heidari-Kaydan, A., Hajidavalloo, E. & Mehrzad, S., (2021). Three-Dimensional Simulation of Leakages in Rotary Air Preheater of Steam Power Plant. *Heat Transfer Engineering*, 1-17. doi:10.1080/01457632.2021.1976563
- Heinlein, A., Klawonn, A., Lanser, M. & Weber, J., (2021). Combining machine learning and domain decomposition methods for the solution of partial differential equations—A review. *GAMM-Mitteilungen*. doi:10.1002/gamm.202100001
- Huang F-L, Sun Z-J, Chen R, Li P-C, Gu J-F, Hu Y-C, & Cheng G. (2015). The investigation of heat transfer model of regenerative air preheaters adapt to SCR denitrification. *Journal of Engineering Thermophysics*, 2683-2688.
- Jagtap, A.D., Kharazmi, E. & Karniadakis, G.E. (2020). Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*. doi:https://doi.org/10.1016/j.cma.2020.113028
- Jagtap, A.D. & Karniadakis, G.E., (2020). Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5), 2002-2041.
- Lagaris, I.E., Likas, A. & Fotiadis, D.I., (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 987-1000. doi:10.1109/72.712178
- Li, C.-H. (1983). A Numerical Finite Difference Method for Performance Evaluation of a Periodic-Flow Heat Exchanger. *J. Heat Transfer*, 611-617. doi:10.1115/1.3245629
- Menasha, J., Dunn-Rankin, D., Muzio, L. & Stallings, J., (2011). Ammonium bisulfate formation temperature in a bench-scale single-channel air preheater. *Fuel*, 2445-2453. doi:https://doi.org/10.1016/j.fuel.2011.03.006
- Moseley, B., Markham, A. & Nissen-Meyer, T., (2021). Finite Basis Physics-Informed Neural Networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. *arXiv preprint arXiv:2107.07871*.
- Muzio, L., Bogseth, S., Himes, R., Chien, Y.C. & Dunn-Rankin, D., (2017). Ammonium bisulfate formation and reduced load SCR operation. *Fuel*, 206, 180-189. doi:https://doi.org/10.1016/j.fuel.2017.05.081
- Penwarden, M., Zhe, S., Narayan, A. & Kirby, R.M., (2021). Physics-Informed Neural Networks (PINNs) for Parameterized PDEs: A Metalearning Approach. *arXiv preprint arXiv:2110.13361*.
- Raissi, M., Perdikaris, P. & Karniadakis, G.E., (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations.

Journal of Computational Physics, 686-707. doi:<https://doi.org/10.1016/j.jcp.2018.10.045>

- Sánchez, J.M.G., Jörgensen, N., Törngren, M., Inam, R., Berezovskyi, A., Feng, L., Fersman, E., Ramli, M.R. & Tan, K., (2021). Edge computing for cyber-physical systems: A systematic mapping study emphasizing trustworthiness. *arXiv preprint arXiv:2112.00619*.
- Shukla, K., Jagtap, A.D. & Karniadakis, G.E., (2021). Parallel physics-informed neural networks via domain decomposition. *Journal of Computational Physics*, 447. doi:<https://doi.org/10.1016/j.jcp.2021.110683>
- Skipeco, T. (1988). The effect of matrix longitudinal heat conduction on the temperature fields in the rotary heat exchanger. *International Journal of Heat and Mass Transfer*, 31(11), 2227-2238. doi:10.1016/0017-9310(88)90155-X
- Sundar, S., Rajagopal, M.C., Zhao, H., Kuntumalla, G., Meng, Y., Chang, H.C., Shao, C., Ferreira, P., Miljkovic, N., Sinha, S. & Salapaka, S. (2020). Fouling modeling and prediction approach for heat exchangers using deep learning. *International Journal of Heat and Mass Transfer*. doi:<https://doi.org/10.1016/j.ijheatmasstransfer.2020.12.0112>
- Wang, H., Planas, R., Chandramowlishwaran, A. & Bostanabad, R., (2021). Train once and use forever: Solving boundary value problems in unseen domains with pre-trained deep learning models. *arXiv e-prints, arXiv--2104*.
- Wang, L., Bu, Y., Li, D., Tang, C. & Che, D., (2019). Single and multi-objective optimizations of rotary regenerative air preheater for coal-fired power plant considering the ammonium bisulfate deposition. *International Journal of Thermal Sciences*, 136, 52-59. doi:<https://doi.org/10.1016/j.ijthermalsci.2018.10.005>
- Zhou, C., Zhang, L., Deng, Y. & Ma, S.C., (2016). Research progress on ammonium bisulfate formation and control in the process of selective catalytic reduction. *Environmental Progress & Sustainable Energy*, 1664-1672. doi:<https://doi.org/10.1002/ep.12409>

BIOGRAPHIES



Vishal Jadhav has completed his Master of Technology degree in Aerospace Engineering from Aerospace Department, IIT Madras, India in 2017. He is currently working as a Scientist at TCS Research with a focus on digital twin development for manufacturing and engineering domains.

He has more than 7 years of experience in applying machine and deep learning, computational fluid dynamics. He has filed several patents and has published research papers in reputed conference proceedings. His current research includes

combining physics and data based algorithms for industrial digital twins



Anirudh Deodhar is currently working as a scientist in TCS Research with a focus on manufacturing and engineering domains. He is a mechanical engineer and holds a master's degree from University of Cincinnati, USA. He has more than 10 years of experience in applying process modeling, computational fluid dynamics, machine/deep learning for industrial digital twins. He is a member of American Society of Mechanical Engineers (ASME). He has published in several international journals and conferences and has filed more than 15 patents. His current research includes fusing physics and deep learning for industrial cyber physical systems.



Ashit Gupta has completed his Master of Technology degree in Aerodynamics from Aerospace Department, IIT Bombay, Mumbai, India in 2017. He has been working as a researcher at Tata Research Development & Design Centre for the past 4.5 years. His expertise lies in machine

learning & deep learning and has contributed towards research for process and manufacturing industries. He has filed several patents and has published research papers in reputed journals and conference proceedings (IEEE, ASME etc.). He has provided solutions like long-term forecasts of health conditions for APH, optimization of operation of boilers in thermal power plants, performance prediction in rotary kiln etc. His interest lies in physics informed deep learning for industrial equipment.



Dr Venkataramana Runkana is currently the Chief Scientist and Head of the Research & Innovation Programme for Manufacturing & Engineering in TCS. He is a chemical engineer by education and holds a Ph.D. from Columbia University, New York. He has more than 30 years of

experience in process modeling, simulation and optimization, advanced data analytics and digital twins, process development, scale-up and design, nanomaterials, and drug delivery systems. He received the TCS Distinguished Scientist Award in 2014 and was an AICTE-INAE Distinguished Visiting Professor at IIT Kanpur during 2013-2018.

APPENDIX

A.1 APH Design and Operational data for base case PINN

Parameter	Values
Matrix	
Inner radius (m)	1.63
Outer radius (m)	8.21
Height of matrix (m)	2.05
Sector Angles	
β_g	180°
β_{a1}	70°
β_{a2}	110°
Material Properties	
Thermal conductivity (W/mK)	52.92
Heat capacity (J/kgK)	456
Gas	
Inlet temperature (°C)	359
Flow rate (kg/s)	770
Primary Air	
Inlet temperature (°C)	25
Flow rate (kg/s)	452.53
Secondary Air	
Inlet temperature (°C)	25
Flow rate (kg/s)	268.13

A Health Index Framework for Condition Monitoring and Health Prediction

Alexander Athanasios Kamtsiuris¹, Florian Raddatz², Gerko Wende³

^{1,2,3} *German Aerospace Center (DLR) Institute of Maintenance, Repair and Overhaul, Hamburg, 21129, Germany*
alexander.kamtsiuris@dlr.de
florian.raddatz@dlr.de
gerko.wende@dlr.de

ABSTRACT

In the field of Maintenance, Repair and Overhaul (MRO), stakeholders such as operators or service providers have to keep track of the health status of fleets of complex systems. The ability to estimate the future health status of these systems and their components becomes more pivotal when seeking to efficiently operate and maintain these systems. Today, these stakeholders have access to a lot of different data sources regarding fleet, operation schedule, ambient condition, system and component information. Many different prognostic methods from different disciplines are available and will further improve henceforward. In many cases these data sources and methods function as isolated methods in their own field. This fragmentation makes a holistic prognosis very challenging in many cases. Therefore, stakeholders need information integrating methods and tools to gain an exhaustive insight into the health status development of the complex assets they are operating or maintaining, in order to make well-founded decisions regarding operation or maintenance planning. In this paper, a Python-based health index framework is presented. It enables users to integrate operation schedules of different detail levels with enriching data sources such as ambient condition data. Furthermore, it provides methods to design complex asset systems which are linked via their construction, function or degradation mechanisms/health indices via transfer relations. It allows to monitor the asset's condition based on operation data and to simulate different operation scenarios regarding the health index development.

1. INTRODUCTION

System Health Management (SHM) plays a key role in today's Maintenance, Repair and Overhaul (MRO) activities by making the asset's operation economically more efficient and

economically competitive. Many Health Management approaches need the development of the health status development according to individual mission profiles of the asset as input for their simulation. Examples are the estimation of operating costs as in (Pohya, Wehrspohn, Meissner, & Wicke, 2021) or deriving prescriptive maintenance strategies as in (Meissner, Rahn, & Wicke, 2021).

A key functionality of Digital Twins of systems is the ability to simulate future health development (Meyer et al., 2020). Many different methods to predict the Remaining Useful Life (RUL) of components have been developed so far (van Nguyen et al., 2019). In practice, for system designers and operators, the system-level prognostics to predict the System Remaining Useful Life (SRUL) are needed since the degradation of the different system components influence each other and hold an additional potential for uncertainty (Tamssaouet, Nguyen, Medjaher, & Orchard, 2021). Often, the operating condition or environmental conditions are factored into the RUL prediction as sources of uncertainty. Moreover, the RUL prediction is carried out based on the individual history of the concerning component or system. In order to allow more precise health prediction and to improve the versatility of decision makers, considering the impact of specific future operating settings and environmental conditions in RUL predictions has gained interest. (Chang, Lin, & Zio, 2022)

In this paper, a generic framework to integrate functional and hardware related information with diagnostic and prognostic methods is proposed. It allows to estimate the current as well as to predict the future health state on a system level, expressed by health indexes, based on operating and environmental conditions. It provides a method to analyze the interdependence of different degradation mechanisms on the system health state and the SRUL. The results can be used for health management activities. The framework is developed in Python and accessible via <https://github.com/DLR-MO/system-health-index-framework>.

The main aspects of the framework are described in section 2. Using the new Commercial Modular Aero-Propulsion Sys-

Alexander Kamtsiuris et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

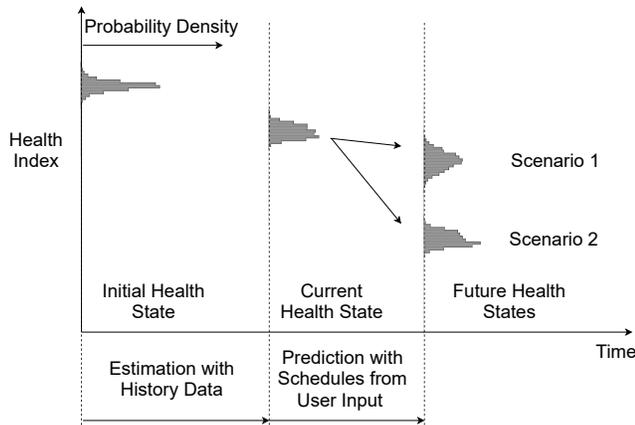


Figure 1. Current and Future Health State

tem Simulation (N-CMAPSS) data presented in (Arias Chao, Kulkarni, Goebel, & Fink, 2021), key functionalities of the framework are demonstrated in section 3. Further development of the framework is discussed in the section 4.

2. ASPECTS OF THE SYSTEM HEALTH INDEX FRAMEWORK

The proposed framework allows to integrate health state estimation and health prediction methods in order to monitor health condition and to simulate the future health status of an arbitrary complex system based on user-defined schedules of ambient and operating conditions, as shown in Figure 1. The health state of a component or system is expressed with the health index. Starting with the current health state the framework allows to simulate different scenarios regarding future operating schedules. In the following subsections different aspects of the framework will be discussed in further detail.

2.1. System Health Index

A system is a hierarchically organized group of subsystems, components or parts which are constructively and functionally related (Kossiakoff, Biemer, Seymour, & Flanigan, 2020). Accordingly, also the different health states interfere with each other. The health state of each of the above mentioned objects can be described by a health index. The health index incorporates multiple, sometimes hidden and not observable degradation processes of different components and therefore allows an analysis of the current and future health states of systems (Sun, Zuo, Wang, & Pecht, 2012).

In this work the definition for the health index proposed in (Arias Chao et al., 2021) is used. The health index hi is defined as in Equation (1). Parameter values of a measurable real or virtual sensor are used and normalized with known reference values for the respective parameter, e.g. known temperature thresholds compared to the temperature values when there is no wear $w = 0$. The health index reaches $hi = 0$ if

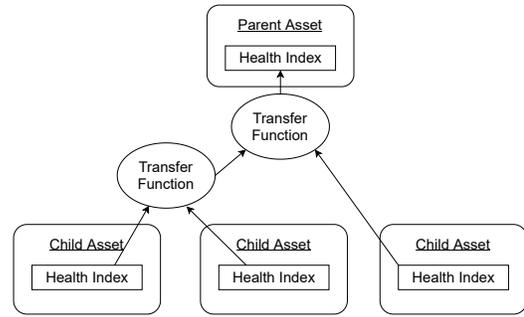


Figure 2. Parent-Child Concept for System Description

the wear w equals the defined threshold wear $w_{threshold}$. The wear depends on time and can also be influenced by operating conditions.

From real and virtual sensor readings and by using health state estimation methods, the history or current health state of a system and its components is estimated. These estimations provide the foundation for the future health state prediction.

$$hi(t) = 1 - \frac{w(t)}{w_{threshold}} \quad (1)$$

Each component has at least one health index for the description of its general health state. The overall health index (HI) of a component can be derived from the set of assigned subordinate health indexes (hi), which is shown in Equation 2. Transfer functions $f_{Transfer}$ express how a certain set of health indexes influences another health index.

$$HI = f_{Transfer}(hi_i) \quad (i = 1, \dots, n) \quad (2)$$

For components in series, usually the minimum health index hi governs the resulting HI . However, for components in parallel, where the failure of one component does not cause a failure of the parent system, the HI corresponds to the maximum hi (Rodrigues et al., 2015). Other transfer functions e.g. functions which apply weights to health indexes, are possible and can be integrated into the presented framework. This is necessary e.g. to describe the health status of an aircraft engine's turbine rotor assembly with cracks on different turbine blades and in different crack propagation states.

This framework uses the parent-child-logic to describe and model systems, as shown in Figure 2.

2.2. Current Health State Estimation and Health State Prediction

In order to predict future health states, it is important to estimate the current health state of a system and its components. The current health state is estimated by using actual and virtual sensor signals. Sensor noise and different operational and environmental conditions have an impact on the sensor

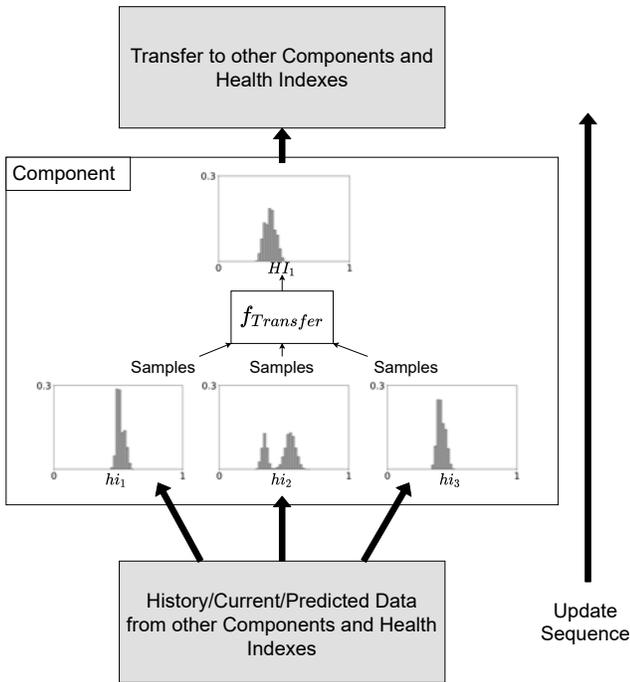


Figure 3. Update Process

measurements, therefore filtering and normalization methods need to be applied in order to get comparable reference values for the sensor parameters independently from the above mentioned factors (Hajiha, Liu, & Hong, 2021). Different promising methods to baseline the sensor data have been developed recently, such as in (Baptista, P. Henriques, & Goebel, 2021) by using neural networks or in (Hou et al., 2021) by introducing sparsity indexes for rotating machines.

Based on the estimated current health state, future health states can be predicted. Among other techniques, health index-based approaches for system health analysis have been found to be effective methods. Various prognostic algorithms such as neural or Bayesian networks have been used. The output can be either the RUL or the offset from normal, healthy states. (Kim, Choi, & Kim, 2021)

This framework uses current and history (virtual) sensor readings and prior health state estimates to calculate the probability distribution for the current health state. Expressing the health state with a probability distribution considers the inherent uncertainty of that estimate. The role of uncertainty is further described in section 2.3.

The health state prediction methods estimate the decrease of the health indices in correlation with an arbitrary set of operational and environmental parameters. The framework allows to integrate prognostic algorithms from different sources. The inputs are the current health state distribution, the prior health state development as well as the user-defined input parameters from the simulation schedules, which are described in more detail in section 2.4. The outputs are increments of

health index consumption per simulation step.

2.3. Uncertainty in Condition Monitoring and Prognostics

Evaluating the uncertainty of estimations plays an important role in condition monitoring and prognostics. (Sankararaman & Goebel, 2015) define four sources of uncertainty in condition-based monitoring and prognostics. Uncertainty management addresses the influence on minimizing uncertainty sources and to administer risk-decreasing measures, such as less present uncertainty by less uncertain inputs from sensors. Uncertainty quantification needs a sensitivity analysis step in order to identify the input parameters which impact the output of a model the most (Razavi et al., 2021).

Present uncertainty describes the uncertainty in estimating the current health state of a system and it is depending on the quality of the sensors and the filtering methods applied. **Future uncertainty** results from the lack of knowledge about future loading, operating, environmental and usage conditions. **Modeling uncertainty** refers to the uncertainties regarding the prediction model, such as the model’s output response on the given inputs of loading, operating, environmental and usage conditions or the model parameters. **Prediction method uncertainty** describes the uncertainty from combining the prior three sources of uncertainty and their impact on the prediction. The Monte Carlo sampling is used most commonly as uncertainty propagation method. Random samples are drawn (such as initial health state, operating conditions, etc.) and the corresponding realizations are computed (e.g. health state after a certain time period). Monte Carlo can be computationally expensive, however, compared to faster uncertainty propagation techniques, it allows to reduce the uncertainty of the estimated probability distribution, the prediction method uncertainty (Sankararaman & Goebel, 2015). This framework uses Monte Carlo sampling for uncertainty propagation, which is incorporated into the update process in Figure 3.

2.4. Discrete Event Schedule

In the context of this framework, schedules are lists of events of arbitrary granularity such as flight, take-off, turn, etc. They consist of historic data and future data. Uncertainty of events, as described in subsection 2.3, have an significant impact on the prediction results. The schedules contain an arbitrary number of parameters used for health state estimation and health state prediction methods. Different techniques to enhance the schedules can be integrated when using the framework.

2.5. Process

1. **Model**
 - (a) **Creating the System**

- Objects of the *Component* class are initialized and linked using the Parent-Child-principle
- Per component a general health index (*HI*) is initialized
- Standard transfer relations between components are initialized

(b) **Defining and Adding Health Indexes to the Model**

- Objects of the *Health_Index* class are defined and added to the model's components
- Per health index (*hi*) start values, reference values, health state estimation method and prediction method are defined
- The transfer target *HI* for the later update process is automatically derived from the component's transfer relations or can be customized by adding an object of the class *Transfer* to the model. This allows to link health indexes with arbitrarily complex transfer functions.

(c) **Adding History Data to the Model**

- A data frame using the principle for discrete event schedules described in subsection 2.4 is created and added to the component's *history* attribute.
- Running the *Component* class' *estimate_current_health_state* function updates the health index values of the model bottom up for every time step in the history data. The scheme for the update process is shown in Figure 3.

2. **Simulation**

(a) **Creating Simulation Schedules**

- The model is loaded and the history data is fetched.
- Using the system's history data, simulation schedules are generated with the information from the user input.

(b) **Running the Simulation**

- The algorithm loops through each simulation schedule and each time step, continuously predicting a degradation increment and updating the system's health state with the update process.
- after the simulation run, each health index progress dependant on the simulation schedules can be analyzed.

3. **CASE STUDY**

In order to demonstrate the functionalities and the process of the framework described in section 2.5 a system consisting of multiple components and various degradation mechanisms respectively health indexes is generated. Afterwards, a set of simulation schedules is created and the simulation is run.

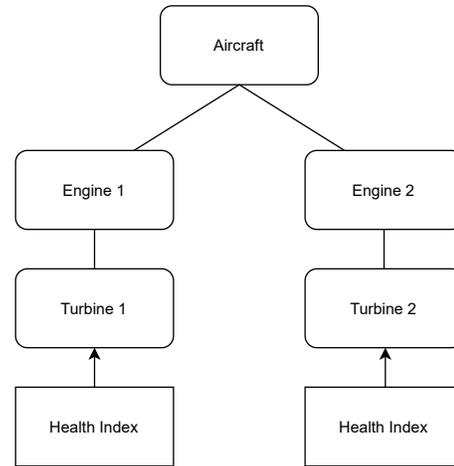


Figure 4. System Example

In a first step, an aircraft model which consists of two engines and respectively two High Pressure Turbine (HPT) modules is set up. For each HPT module, the health index based on the measured total temperature at the HPT outlet is established. Therefore, this health index incorporates not only the wear of the component, which it is directly linked to, but the wear of all components which have an impact on the reference sensor measurement. As transfer functions for the health indexes *min()* is chosen, since due to safety obligations, the unhealthiest turbine dictates the overall system's health status. The model is depicted in Figure 4.

For the introduced health indexes, health estimation and prediction methods need to be defined.

For demonstration purposes, data from (Arias Chao et al., 2021), which is created with the Commercial Modular Aero-Propulsion System Simulation (CMAPSS) model, described in (Frederick, DeCastro, & Litt, 2007), is used. The data set N-CMAPSS contains engine run-to-failure data for a fleet of aircraft engine units under real flight conditions. A method for the current health state estimation and the future health prediction is derived by using the data set *DS01*. Generally, the system's degradation in (Arias Chao et al., 2021) is induced by modifying the engine health parameters flow and efficiency of different engine modules. In practice, these parameters are not measured, however, the degradation is indicated by observable sensor measurements such as temperature measurements in the turbine.

In this case study, the method used in (Arias Chao et al., 2021) for engine health parameter alteration is transferred to the health index decrease in order to model its degradation over time. The initial wear δ_0 is obtained from a uniform distribution. The first phase of a normal, linear degradation until time t_s is modeled with a constant slope a_n and with t being the flight cycles.

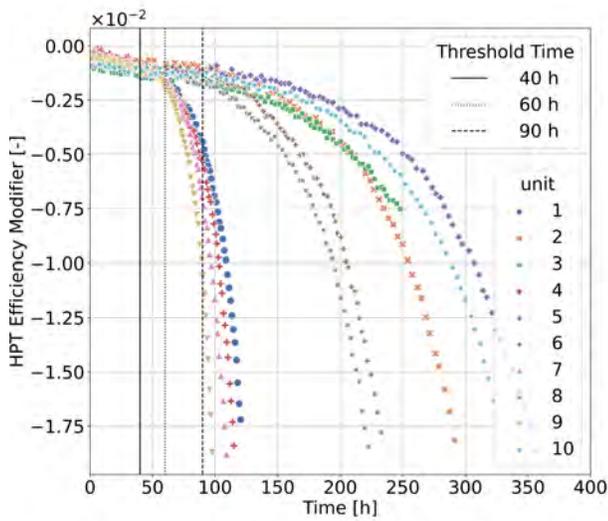


Figure 5. Development of High Pressure Turbine Efficiency Modifier Over Time from the N-CMAPSS Data Set DS01

$$\delta_n(t) = a_n t + \delta_0 \quad \forall t < t_s \quad (3)$$

t_s is the time when the excitation energy exceeds the maximum excitation energy E_{max} of a component for the first time.

$$t_s = \inf\{t \geq 0 \mid E(t) > E_{max}\} \quad (4)$$

The excitation energy is the integral of the power consumed and produced by a component over a certain time interval. In case of Equation 5, t denotes the time in hours.

$$E(t) = \int_0^t P(t) dt \quad (5)$$

Between equal subsystems, the maximum excitation energy E_{max} varies due to the individual material properties. Once t_s is reached, the abnormal degradation δ_a is described by the following model:

$$\delta_a = 1 - e^{a(t-t_s)^b} + \delta_n(t_s) + \xi \quad (6)$$

a and b are uniformly distributed parameters for the exponential function, t is the number of cycles and ξ is the process noise, which covers the uncertainty e.g. originating from sources such as maintenance events, and therefore can be either positive or negative.

The degradation in the data set *DS01* is governed by a decrease of the HPT efficiency.

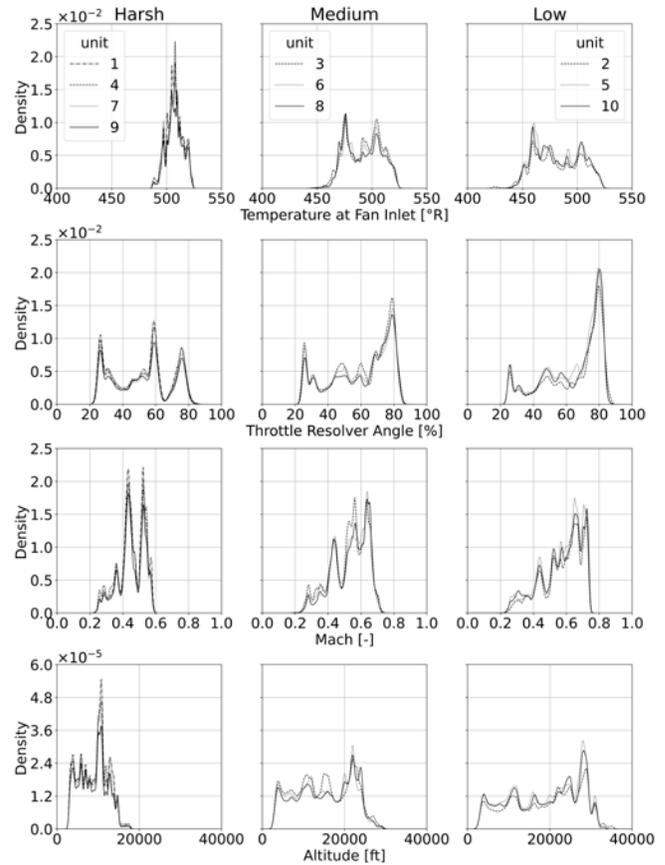


Figure 6. Operation Settings from N-CMAPSS data set *DS01*

Figure 5 shows the decrease of the HPT efficiency modifier over the flight hours collected by each unit. As input variables for the flights, the operation settings depicted in Figure 6 are used by the original authors. The throttle resolver angle (TRA) describes the used relative power setting for the engine. The onset of abnormal degradation marks the hours collected when E_{max} is reached. Roughly, there are three characteristic groups of flight trajectories regarding the t_s , where both similar onset of abnormal degradation and similarity in operating settings can be observed, see Table 1 and Figure 6. The groups are called *Harsh*, *Medium* and *Low* in this work, related to each group's characteristic onset of abnormal degradation and the respective excitation power P_{rel} . Unit 3 differs from that behaviour, since the operating settings resemble the *Medium* class flights, whereas the trajectory is closer to the *Low* class. A possible explanation might be the method, how variable material properties are incorporated in the synthesis of the data in the original work of (Arias Chao et al., 2021). The individual maximum excitation energy of a component is modelled by a Gaussian distribution. With a higher maximum excitation energy, the shift from linear to abnormal degradation is experienced later, even with relatively harsher operating conditions.

Table 1. Flight Intensity Classes for Schedule Generation

Intensity Class	Units	$t_{threshold}$ [h]	P_{rel} [1/h]
Harsh	1,4,7,9	40	-0.025
Medium	3,6,8	60	-0.017
Low	2,5,10	90	-0.011

Table 2. Simulation Parameters

	Harsh	Low
$cycle\ length$ [h]	$\mathcal{N}(1.2, 0.2)$	$\mathcal{N}(4, 0.75)$
P_{rel} [1/h]	-0.025	-0.011
a	$\mathcal{U}(0.001, 0.003)$	$\mathcal{U}(0.001, 0.003)$
b	$\mathcal{U}(1.4, 1.6)$	$\mathcal{U}(1.4, 1.6)$
ξ	$\mathcal{N}(0, 0.001)$	$\mathcal{N}(0, 0.001)$
$n_{SimulationRuns}$	100	100
$n_{HistoryCycles}$	20	20
a_n	-0.001	-0.001
δ_0	$\mathcal{U}(0.9, 1.)$	$\mathcal{U}(0.9, 1.)$
Transfer Function	$min()$	$min()$

Harsh conditions have relatively higher temperatures at the fan inlet compared to low intensity operation conditions. The inlet temperature seems to have a higher impact on increasing the excitation energy level than the averagely higher TRA observed in the *Low* group.

The relative excitation power is calculated with Equation 7.

$$P_{rel} = \frac{1}{t_{threshold}} \quad (7)$$

For demonstration purpose, the equations 3 - 7 are used for both the health state estimation as well as for the health state prediction in this case study.

Event schedules described in subsection 2.4 are generated and the health state development for different operating and environmental conditions is simulated. The current health state is estimated starting from the beginning of life of the system until the time point $t_{current} = 20\ cycles$ using *harsh* condition setting, which implies a excitation power of $P_{rel} = -0.025\ [1/h]$. For each of the two operation intensity classes *harsh* and *low*, a simulation with $n_{SimulationRuns} = 100$ schedules is set up. The two scenarios have different excitation power parameters P_{rel} and different $cyclelength$ parameters, an overview of the used

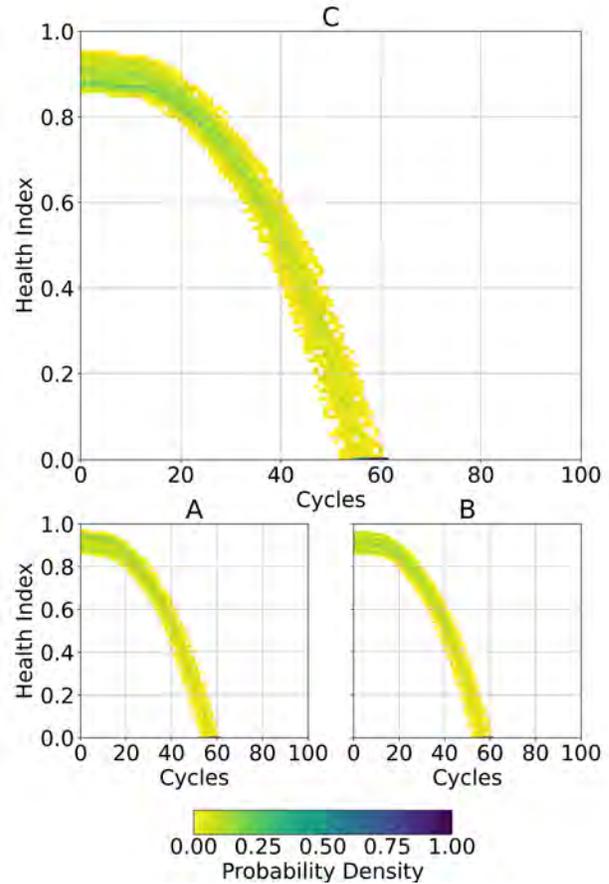


Figure 7. Prediction for Harsh Intensity Operating Conditions. A: *hi* EGT Turbine 1, B: *hi* EGT Turbine 2, C: *HI* Aircraft

parameters is given in Table 2.

The prediction for the simulation of future harsh operating conditions is depicted in Figure 7. Around 15 cycles into the simulation the abnormal degradation is reached. SRUL for the Aircraft with probability $p = 1$ is at 61 cycles.

The distributions of the turbine health indexes in Figure 7 A and B show an increasing spread of possible health indexes at each time step. Towards the end the spread then decreases due to the probability p of $hi = 0$ reaching $p = 1$. The applied transfer function $min()$ causes a shift to the lower health indexes for the aircraft level health index in Figure 7 C.

The prediction for the simulation of future low intensity operating conditions is depicted in Figure 8. Even though the intensity is lower than in scenario 1, SRUL with probability $p = 1$ is reached already at 55 cycles. This is due to the higher $cycle\ length$ in scenario 2, which leads to an earlier excess of the excitation energy level E_{max} and therefore an

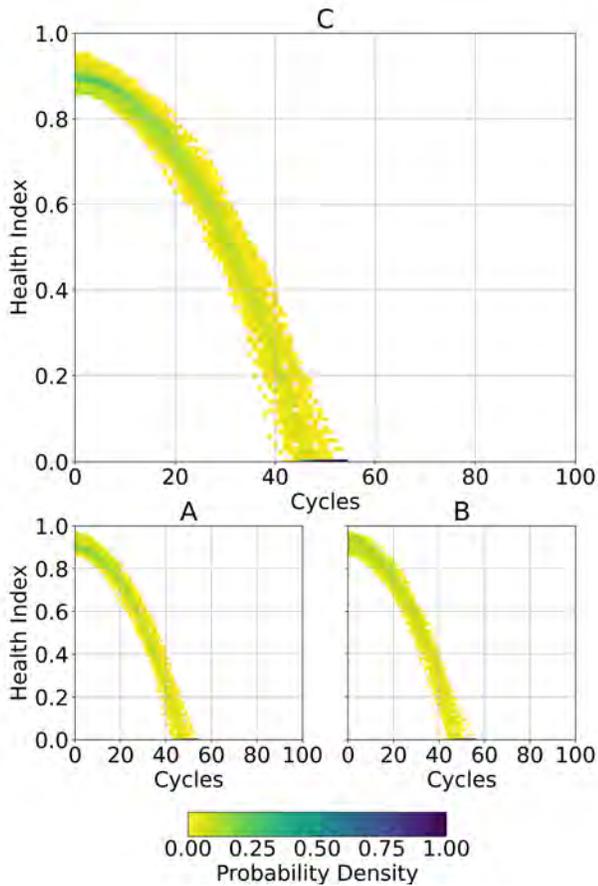


Figure 8. Prediction for Low Intensity Operating Conditions. A: *hi* EGT Turbine 1, B: *hi* EGT Turbine 2, C: *HI* Aircraft

earlier onset of abnormal degradation. Turbine 1 with averagely lower health indexes dictates the overall system health index of the aircraft, especially in the first cycles.

In both scenarios, with progressing time, the spread of the health index increases until the first simulations reach $hi = 0$. The possibility of using different transfer functions shows, that more sophisticated dependency between health index degradation trajectories can be established, which allows more accurate decision making for the user.

4. CONCLUSION

The proposed framework establishes a method to integrate health state estimation and health state prediction for complex systems. It allows to create system models using the parent-child principle and to add health indexes for different degradation mechanisms. These health indexes can be linked via transfer functions. The framework combines history data with future event schedules to simulate future health states. It takes into account the uncertainty propagation by

using Monte-Carlo-Sampling. The framework’s output is an important input for further health management activities.

In the future, the integration of online and offline prognostic metrics in order to assess the impact of the diagnosis and prognostic algorithms on the uncertainty will be investigated. Furthermore, uncertainty management functionalities such as sensitivity analysis in order to analyse the impact of single factors on the prediction uncertainty will be assessed, e.g. for sensor improvement activities.

Moreover the establishment of interdependence between health index developments of different hierarchical levels will be improved. Also, other uncertainty propagation techniques to improve the performance of the simulation will be investigated.

ACKNOWLEDGEMENT

The research associated with this paper was performed within the project Digital Twin for Engine, Components and Aircraft Technologies (DigECAT), which is the 2nd phase of the digital twin project within the aviation program of the German Aerospace Center (DLR).

REFERENCES

- Arias Chao, M., Kulkarni, C., Goebel, K., & Fink, O. (2021). Aircraft engine run-to-failure dataset under real flight conditions for prognostics and diagnostics. *Data*, 6(1), 5. doi: 10.3390/data6010005
- Baptista, M. L., P. Henriques, E. M., & Goebel, K. (2021). A self-organizing map and a normalizing multi-layer perceptron approach to baselining in prognostics under dynamic regimes. *Neurocomputing*, 456, 268–287. doi: 10.1016/j.neucom.2021.05.031
- Chang, L., Lin, Y.-H., & Zio, E. (2022). Remaining useful life prediction for complex systems considering varying future operational conditions. *Quality and Reliability Engineering International*, 38(1), 516–531. doi: 10.1002/qre.2997
- Frederick, D. K., DeCastro, J. A., & Litt, J. S. (2007). *User’s guide for the commercial modular aero-propulsion system simulation (c-mapss)* (Tech. Rep. No. E-16205). Washington DC USA: NASA.
- Hajiha, M., Liu, X., & Hong, Y. (2021). Degradation under dynamic operating conditions: Modeling, competing processes and applications. *Journal of Quality Technology*, 53(4), 347–368. doi: 10.1080/00224065.2020.1757390
- Hou, B., Wang, D., Xia, T., Wang, Y., Zhao, Y., & Tsui, K.-L. (2021). Investigations on quasi-arithmetic means for machine condition monitoring. *Mechanical Systems and Signal Processing*, 151, 107451. doi: 10.1016/j.ymssp.2020.107451
- Kim, S., Choi, J.-H., & Kim, N. H. (2021). Challenges

- and opportunities of system-level prognostics. *Sensors (Basel, Switzerland)*, 21(22). doi: 10.3390/s21227655
- Kossiakoff, A., Biemer, S. M., Seymour, S. J., & Flanigan, D. A. (2020). *Systems engineering: Principles and practice* (Third edition ed.). Hoboken, NJ: John Wiley & Sons Inc. doi: 10.1002/9781119516699
- Meissner, R., Rahn, A., & Wicke, K. (2021). Developing prescriptive maintenance strategies in the aviation industry based on a discrete-event simulation framework for post-prognostics decision making. *Reliability Engineering & System Safety*, 214, 107812. doi: 10.1016/j.res.2021.107812
- Meyer, H., Zimdahl, J., Kamtsiuris, A., Meissner, R., Radatz, F., Haufe, S., & Bäbler, M. (2020). *Development of a digital twin for aviation research*. Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V. doi: 10.25967/530329
- Pohya, A. A., Wehrspohn, J., Meissner, R., & Wicke, K. (2021). A modular framework for the life cycle based evaluation of aircraft technologies, maintenance strategies, and operational decision making using discrete event simulation. *Aerospace*, 8(7), 187. doi: 10.3390/aerospace8070187
- Razavi, S., Jakeman, A., Saltelli, A., Prieur, C., Iooss, B., Boronovo, E., ... Maier, H. R. (2021). The future of sensitivity analysis: An essential discipline for systems modeling and policy support. *Environmental Modelling & Software*, 137, 104954. doi: 10.1016/j.envsoft.2020.104954
- Rodrigues, L. R., Gomes, J. P. P., Ferri, F. A. S., Medeiros, I. P., Galvao, R. K. H., & Nascimento Junior, C. L. (2015). Use of phm information and system architecture for optimized aircraft maintenance planning. *IEEE Systems Journal*, 9(4), 1197–1207. doi: 10.1109/JSYST.2014.2343752
- Sankararaman, S., & Goebel, K. (2015). Towards characterizing the variability in the loading demands of an unmanned aerial vehicle. In *17th aiaa non-deterministic approaches conference 2015*. Red Hook, NY: Curran. doi: 10.2514/6.2015-1597
- Sun, J., Zuo, H., Wang, W., & Pecht, M. G. (2012). Application of a state space modeling technique to system prognostics based on a health index for condition-based maintenance. *Mechanical Systems and Signal Processing*, 28, 585–596. doi: 10.1016/j.ymsp.2011.09.029
- Tamssaouet, F., Nguyen, K. T. P., Medjaher, K., & Orchard, M. (2021). Fresh new look for system-level prognostics. *International Journal of Prognostics and Health Management*, 12(2). doi: 10.36001/ijphm.2021.v12i2.2777
- van Nguyen, D., Kefalas, M., Yang, K., Apostolidis, A., Olhofer, M., Limmer, S., & Bäck, T. (2019). A review: Prognostics and health management in automotive and aerospace. *International Journal of Prognostics and Health Management*, 10(2). doi: 10.36001/ijphm.2019.v10i2.2730

Tool Compatibility Index: Indicator Enables Improved Tool Selection for Well Construction

Jinlong Kang¹, Christophe Varnier², Ahmed Mosallam³, Noureddine Zerhouni⁴, Fares Ben Youssef⁵, and Nannan Shen⁶,

^{1,2,4} *AS2M, FEMTO-ST, ENSMM, UBFC, Besançon, 25000, France*

jinlong.kang@femto-st.fr

christophe.varnier@femto-st.fr

noureddine.zerhouni@femto-st.fr

^{1,3,6} *Schlumberger, Clamart, 92140, France*

JKang5@slb.com

AMosallam@slb.com

NShen@slb.com

⁵ *Schlumberger, Youngsville, 70592, United States*

FYoussef@slb.com

ABSTRACT

In the area of well construction, the tool reliability and the field environment are two contributing factors that influence drilling job efficiency and success. Either using high specification tools in low-risk environmental or applying tools of low reliability in harsh environments is inadvisable. Thus, how to select a suitable tool fitting the environment of an approaching drilling job is of great significance for tool planning. However, today, the tool selection decision is not optimized because it is often based on partial data availability and understanding.

This paper presents an indicator called tool compatibility index, which can support improved tool selection decision making. This index takes part reliability, part criticality, and field environment into consideration, and gives a score indicating the compatibility of the tool to a specific environment. Moreover, the tool compatibility index is computed based on a weighted average method, which is computation simple and can be easily deployed. This work is part of a long-term project aiming to construct a risk based decision advisor for drilling and measurement tools.

1. INTRODUCTION

The drilling system (shown in Fig. 1) in the oil and gas industry is usually consisting of a drilling rig, drillpipe, a bottom

hole assembly (BHA) and a drill bit. The BHA is an important part of the drilling system because it must provide power for the bit to rotate and break through the rock, survive a harsh operating environment, and provide accurate directional control of the well (Schlumberger, 2022). The BHA is configured based on drilling operation requirements; thus, different drilling jobs could have different BHA configurations. Nevertheless, the BHA frequently includes measurement-while-drilling (MWD) tool(s), logging-while-drilling (LWD) tool(s), and rotary steering system(s) as shown in Fig. 2. The MWD tool on the top of the BHA is responsible for delivering real-time data to the surface, powering and transmitting data from multiple LWD tools, and determining the position and orientation of the drillstring (Schlumberger, 2022). The LWD combines a complete set of functions, including formation evaluation, well placement, and drilling optimization measurements into a single collar (Mosallam, Laval, Youssef, Fulton, & Viassolo, 2018). The rotary steering system at the bottom of the BHA is designed to rotate the drill bit in the desired direction; thereby, control the well path (Kirschbaum et al., 2020). MWD, LWD, and rotary steering system are collectively termed drilling and measurement (D&M) tools or technologies.

Each D&M tool is an electronics-rich system and through decades of development, the built-in electronic boards have had various design revisions, which cause the reliabilities of the boards to be different. This in turn affects the overall tool reliability. Meanwhile, the reliability of D&M tools plays an important role in drilling operation (Kale, Carter-Journet, Falgout, Heuermann-Kuehn, & Zurcher, 2014). Another main

Jinlong Kang et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

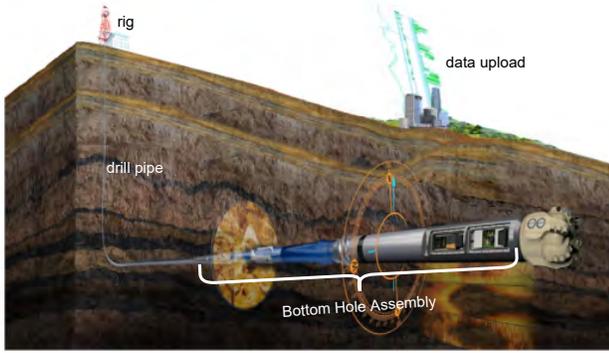


Figure 1. Drilling system schematic.



Figure 2. Bottom hold assembly schematic.

factor affecting the success of a drilling job is the field environment because operating conditions could have a great impact on tool reliability. For example, elevated vibration can cause mechanical structure damage, and high temperature and humidity can cause electronic malfunctions (Bhargava et al., 2020). For an upcoming drilling job, if the D&M tools that make up the BHA are not compatible with the field environment, this would cause the job to fail and/or tool failures, resulting in a huge economic loss. From the perspective of field team, they definitely would prefer to use the most reliable D&M tools to configure the BHA. In this way, the BHA reliability can be maximized, but this is unrealistic and nonoptimal. On the one hand, the availability of tools of high reliability is limited. If these tools were used for low-environmental-risk drilling jobs, then there might not be suitable tools for high-risk drilling jobs. On the other hand, tools of low reliability might be not compatible with harsh environments but could be suitable for drilling operations in moderate environments. In addition, tools of higher reliability or configuration generally means added manufacturing cost. Therefore, selecting the correct D&M tool for the correct field environment is of great significance for tool planning and cost savings.

Unfortunately, today's the tool selection decision is not fully centralized because it often relies on partial well parameters (e.g., hole size) and the technician's understanding of the tool. In addition, the tool selection is usually requires the technician to manually check many datasheets, a labor intensive and inefficient process. We will go into more details about the current tool selection in next section. Considering these mentioned challenges, we propose a new indicator to characterize

the compatibility or fitness of tools vs. different field environments. To the best of our knowledge, this paper is the first to study the D&M tool selection in oil and gas industry.

The remainder of this paper is organized into four sections. The first section presents extra information about the research problem, current tool selection processes, and limitations. The second section presents the proposed solution in detail. The following section presents application scenarios using actual data to confirm the solution. The final section summarizes and proposes some research directions for the future.

2. PROBLEM FORMULATION

In this section a brief introduction of the digital fleet management system (DFMS) used in current tool selection decision-making is presented. Then, a detailed description about current tool selection processes and limitations are presented. At the conclusion of this section, we clarify the research problem presented in this paper.

2.1. What is DFMS?

The DFMS is a commercialized business information dashboard to help field teams choose the most reliable D&M tool for a given job. As mentioned, the revision design of D&M tool electronic boards develops with time. Even with the same design, the electronic components can slightly differ from batch to batch. Indeed, no two tools today have the same hardware configuration or reliability. The DFMS is designed to extract equipment quality and tracking system data for all boards, and compute a reliability score based on the following:

- Manufacturing process changes
- Supplier traceability
- Design changes

The reliability score has three levels; i.e., Level 1 meaning the least reliable and Level 3 indicating the most reliable.

The DFMS output is a dashboard containing the configurations, e.g., equipment hierarchy, parts status (e.g., active, junked, and lost in hole), and parts identity information (i.e., part number, serial number), parts revision and parts reliability scores of all active D&M tools for each location. The two words, part and board are used interchangeably in this paper unless otherwise specified.

2.2. Current Tool Selection Process and Limitations

Currently, the tool selection decision making involves the following three steps as shown in Fig. 3:

1. The field engineer obtains some general parameters (e.g., geographical coordinate, temperature, flow rate, and hole

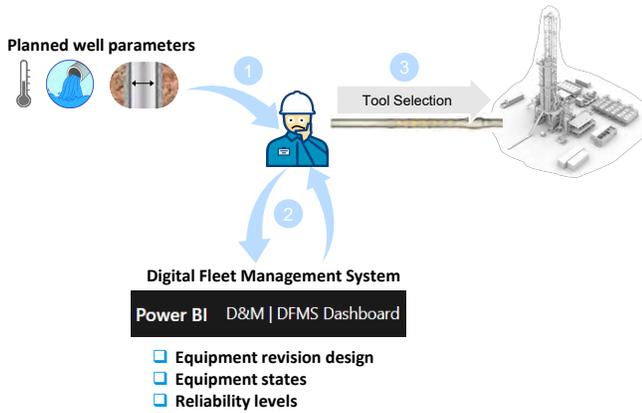


Figure 3. Current tool selection process.

size) of the planned well where the upcoming drilling job will take place.

2. The field engineer refers to the DFMS where information is available about equipment revision design, equipment states and parts reliability levels of tools in the location.
3. According to the DFMS output and experience, the field engineer decides which tool to deploy.

Although the DFMS provides the parts reliability levels profile, the parts reliability levels do not contain environmental definitions or hold systematic criticality information. That is, field environment and parts importance are not considered when computing the reliability scores. Thus, the DFMS output only reveals which tool or part is more reliable, it does not quantify the compatibility of the tool with respect to the field environment. This makes current tool selection decision making rely greatly on empirical knowledge. As mentioned in Section 1, the current tool selection is also labor intensive. In order to achieve objective, effective, and efficient tool selection decision making, it was decided to develop a tool compatibility computation method, which can provide the field user scores range from 0% to 100% indicating the comparability of tools under a specific field environment (e.g., high temperature, medium vibration, and shocks) where the upcoming job will demonstrate.

3. PROPOSED SOLUTION

This section will first describe the criticality rules. Then, we will formalize the proposed tool compatibility computation method and demonstrate it with an example. Finally, an overview of the the proposed solution framework will be presented.

3.1. Criticality Rules

The criticality rules are defined by subject matter experts (SMEs). Different tools have different rules. The rules contain parts criticality information and environment definitions, which over-

Table 1. An excerpt from the criticality rules of a specific LWD tool

Part Name	DFMS Reliability Level	Temperature	Lateral Vibration	Lateral Shock	Criticality
X215	1	NA	NA	NA	4
X215	2	1,2	1,2	1,2	4
X215	3	1,2,3	1,2,3	1,2,3	4
X105	1	NA	NA	NA	3
X105	2	1,2	1,2	1,2	3
X105	3	1,2,3	1,2,3	1,2,3	3
SX207	1	1	1	1	2
X207	2	1,2	1,2	1,2	2
X207	3	1,2,3	1,2,3	1,2,3	2
X117	1	1	1	1	1
X117	2	1,2	1,2	1,2	1
X117	3	1,2,3	1,2,3	1,2,3	1

come the disadvantages of the DFMS reliability levels.

Not all parts share the same failure impact. Moreover, different parts have different failure rates. As a result, SMEs define the importance of the parts based on historical service quality statistics. More specifically, the service quality statistics SMEs used here is failure event occurrence. Since the failure events of each part are recorded, SMEs can easily obtain the number of failure events for each part. The criticality of each part can be then determined based on simple binning approaches. For example, one can assign criticality of 1 to those parts whose failure event occurrences are less than or equal to 5, and assign criticality of 2 to those parts whose failure event occurrences are between 6 and 10. SMEs also help to define rules of mapping between parts' reliability levels and critical environments. Here the critical environments mean the contributing environmental factors (e.g., temperature, vibration) that can cause tool failure. Different D&M tools may have different critical environments.

Table 1 shows an excerpt from the criticality rules of a specific LWD tool which are defined by the corresponding SME. In the table, the 'NA' means this part is not fit for any environment, in other words, this part is obsolete. The number '1', '2', and '3' in Temperature column indicating low, medium, and high temperature, respectively. The same applies for the Lateral Vibration and Lateral Shock columns. The Criticality column shows the importance of parts. The larger the value, the more important or critical is the part. For example, the first row of the table indicates X215 parts with reliability level 1 should not used, X215 parts have criticality value of 4. The fifth row suggests that X105 part with reliability level 2 can be used in a field environment of low and medium temperature, low and medium lateral vibration, low and medium lateral shock. X215 parts have criticality values of 3.

DFMS output		Criticality Rules				x_i
Part name	Reliability levels	Temperature	Lateral Vibration	Lateral Shock	Criticality w_i	
X215	3	1,2,3	1,2,3	1,2,3	4	1
X103	2	NA	NA	NA	4	0
X106	3	1,2,3	1,2,3	1,2,3	4	1
X106	3	1,2,3	1,2,3	1,2,3	4	1
X105	2	1,2	1,2	1,2	3	0
X102	2	NA	NA	NA	3	0
X010	3	1,2,3	1,2,3	1,2,3	3	1
PNG	3	1,2,3	1,2,3	1,2,3	2	1
X316	2	1,2	1,2	1,2	2	0
X207	2	1,2	1,2	1,2	2	0
X001	2	1,2	1,2	1,2	2	0
X211	3	1,2,3	1,2,3	1,2,3	2	1
X113	3	1,2,3	1,2,3	1,2,3	1	1
X012	2	1,2	1,2	1,2	1	0
X214	3	1,2,3	1,2,3	1,2,3	1	1
X022	3	1,2,3	1,2,3	1,2,3	1	1
X123	3	1,2,3	1,2,3	1,2,3	1	1
X004	3	1,2,3	1,2,3	1,2,3	1	1
X117	1	1	1	1	1	0
BU201	3	1,2,3	1,2,3	1,2,3	1	1
BD001	3	1,2,3	1,2,3	1,2,3	1	1
X009	3	1,2,3	1,2,3	1,2,3	1	1

$I = 60\%$

Figure 4. Tool compatibility computation example.

3.2. Weighted Average Based Tool Compatibility Index

The proposed tool compatibility index I is mathematically expressed as

$$I = \frac{\sum_{i=1}^N w_i x_i}{\sum_{i=1}^N w_i} \times 100\% \quad (1)$$

where N is the number of parts in the tool, w_i is the criticality value of part i in the tool, x_i is 1 if part i can be used for the specific environment; otherwise, x_i is 0. We can infer the w_i and x_i according to the criticality rules defined by SMEs.

Next an example of an LWD tool shows how this compatibility index is calculated. The example is shown in Fig 4. Suppose the upcoming job is going to be run in an oil field of medium temperature, high lateral vibration, and high lateral shock environment. Then according to the DFMS output and criticality rules, we can infer the x_i of the tool that is shown in the right-hand side of the figure. For example, the X105 part in the fifth row can be used in a medium temperature environment but cannot be used in high lateral vibration or high lateral shock environment according to the criticality rules; thus, the corresponding x_i is 0. After obtaining the x_i of all parts in this tool, plugging both the criticality value w_i and x_i into Eq. (1), we determine the tool compatibility index of this tool under the specified environment is 60%. In addition, The zeros of x_i indicate the noncompatible parts of this tool, which suggest these parts need to be upgraded if the field engineer wants to use this tool for the specified environment.

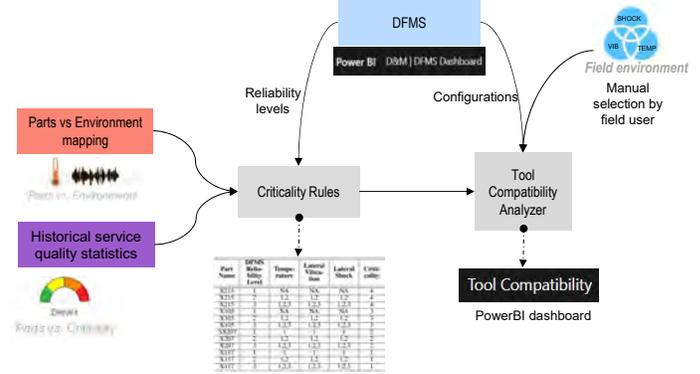


Figure 5. Framework of the proposed solution.

3.3. Framework of The Proposed Solution

Based on previously mentioned criticality rules and compatibility index calculation method, the framework of the new solution for tool selection is shown in Fig. 5. The steps in the framework are described as follows.

1. The SMEs define the criticality rules for D&M tools derived from the mapping of parts vs. environment, historical service quality statistics, as well as DFMS reliability levels.
2. As specified by DFMS, the tool configurations can then be derived. The configurations are combined with the field environment chosen by the field user, and then fed into the tool compatibility analyzer (i.e., weighted average based tool compatibility index).
3. The compatibility index of the tool is outputted. In addition, if needed, noncompatible parts and their upgrade cost information can be acquired.
4. By use of step 2 and 3 for the tool fleet and all field environment combinations (e.g., there are $3^3 = 27$ combinations for three environment categories (e.g., temperature, lateral vibration, lateral shock) of three levels (i.e., low, medium, high)), the compatibility indices of the tool fleet for each environment combination can be achieved. A dashboard can be established using these outputs.

4. USE CASES

The output of the proposed solution is a dashboard that contains compatibility indices of tool fleet vs. different environments. This section presents the use case diagram and two application scenarios to demonstrate the effectiveness of our solution. It should be noted that some information is not included due to confidentiality.

4.1. Use Case Diagram

Figure 6 shows the use case diagram. The dashboard users are field engineers who are responsible for tool operation. The field user selects the field environment of the upcoming job

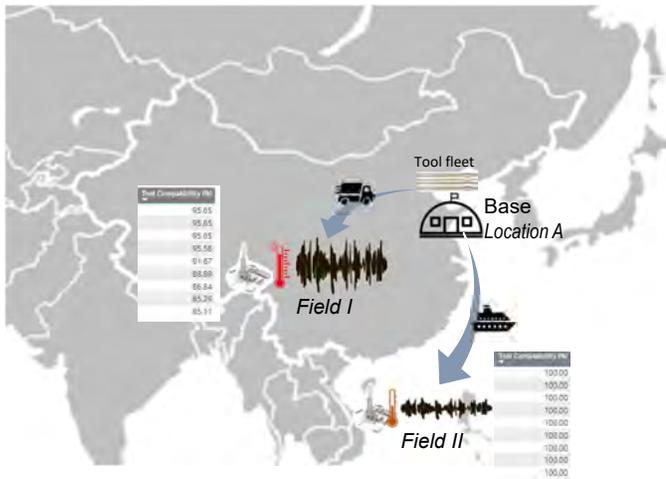
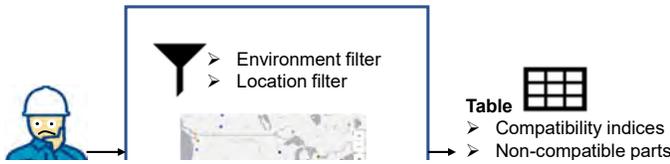


Figure 7. Two application scenarios.

and the user base location using the environment filter and location filter. The dashboard will then output a table containing compatibility indices information of the tool fleet in that location, suggest upgrade plans and costs for noncompatible parts of each tool, and etc.

4.2. Application Scenarios

This subsection presents two scenarios (see Fig. 7) to demonstrate the solution.

Scenario I: A new drilling job was to be operated in *Field I*, a *high temperature, high lateral vibration and shock* environment. The job was assigned to the *Location A* base. The compatibility indices of the tool fleet for this environment are also shown adjacent to ‘Field I’ in Fig. 7.

Scenario II: A new drilling job was planned for *Field II*, a *medium temperature, low lateral vibration and shock* environment. The job was also assigned to the *Location A* base. The compatibility indices of the tool fleet under this environment are shown adjacent to ‘Field II’ in in Fig. 7.

Based on the compatibility indices for the two scenarios, it is

shown that under different environments, the same tool fleet has different tool compatibility indices. The tool compatibility indices of the harsh environment (i.e., Scenario I) are less than the mild environment (i.e., Scenario II). If setting a compatibility index threshold of 90% as tool selection criteria, then only the first five tools are compatible with the harsh environment; i.e., only those tools can be used for the Scenario I drilling job. On the other hand, all of the tools can be selected for the Scenario II drilling job because all of the compatibility indices are greater than 90%.

5. CONCLUSIONS AND FUTURE WORK

A new solution for D&M tool selection in oil and gas industry has been presented in this paper. The proposed solution has two main advantages, that is, it integrates potential environmental risks and part criticality into a tool selection decision making process. The new solution is successfully implemented into a business information platform (i.e., Microsoft Power BI) and verified through field tests. By applying this solution, the tool selection becomes more objective and efficient because some of the manual checking processes are optimized.

Several challenges exist for our solution that will be examined in the future. For example, one major challenge is that the proposed solution needs a user to select the potential field environment of the approaching job based on the user’s domain knowledge. Considering this challenge, the authors developed a dashboard for field environment characterization based on historical tool environmental exposure data, and will study how to link this dashboard to tool compatibility. In this way, user selection of field environment will become needless. Furthermore, the tool compatibility is regardless of part reliability affected by cumulative environmental exposure. Then, considering part cumulative environmental exposure into compatibility index computation is also worthwhile study.

ACKNOWLEDGMENT

This work has been supported by the EIPHI graduate school (contract “ANR-17-EURE-0002”) and the CIFRE fellowship (n°2020/0127). The authors thank Rod Hotz from Schlumberger and the two anonymous reviewers from the conference whose comments helped improve and clarify this manuscript.

REFERENCES

Bhargava, C., Sharma, P. K., Senthilkumar, M., Padmanaban, S., Ramachandaramurthy, V. K., Leonowicz, Z., ... Mitolo, M. (2020). Review of health prognostics and condition monitoring of electronic components. *IEEE Access*, 8, 75163–75183. doi: 10.1109/ACCESS.2020.2989410

Kale, A. A., Carter-Journet, K., Falgout, T. A., Heurmann-

Kuehn, L., & Zurcher, D. (2014). A probabilistic approach for reliability and life prediction of electronics in drilling and evaluation tools. In *Annual conference of the PHM society*.

Kirschbaum, L., Roman, D., Singh, G., Bruns, J., Robu, V., & Flynn, D. (2020). AI-driven maintenance support for downhole tools and electronics operated in dynamic drilling environments. *IEEE Access*, 8, 78683-78701. doi: 10.1109/ACCESS.2020.2990152

Mosallam, A., Laval, L., Youssef, F. B., Fulton, J., & Viasolo, D. (2018). Data-driven fault detection for neutron generator subsystem in multifunction logging-while-drilling service. In *PHM society european conference*.

Schlumberger. (2022). *BHA*. Retrieved 2022-03-25, from <https://glossary.oilfield.slb.com/en/terms/b/bha>

Schlumberger. (2022). *measurements-while-drilling*. Retrieved 2022-03-25, from <https://glossary.oilfield.slb.com/en/terms/m/measurements-while-drilling>

BIOGRAPHIES



Jinlong Kang is currently a PhD student at University of Franche-Comté in Besançon and a data scientist at Schlumberger technology center in Clamart, France. He received the B.S. degree in Industrial Engineering in 2016 and the M.S. degree in Mechanical Engineering in 2019 both from University of Electronic Science and Technology of China.

His main research interests are Prognostic and Health Management (PHM), maintenance decision-making, data mining and machine learning.



Christophe Varnier is an Associate Professor at ENSMM, France. He obtained his Ph.D. degree in 1996 at the University of Franche-Comté in Besançon, France. He is teaching computer science at Ecole Nationale Supérieure de Mécanique et des Microtechniques (ENSMM) since 1996.

He is a Researcher at the Automatic Control and Micro-Mechatronic Systems Department of the FEMTO-ST Institute. His research interests include PHM, operation research, scheduling and optimization.



Ahmed Mosallam is an Analytics Manager at Schlumberger technology center in Clamart, France. He has his Ph.D. degree in automatic control in the field of PHM from University of Franche-Comté in Besançon, France. His main research interests are signal processing, data mining, machine learning and PHM.



Noureddine Zerhouni received his engineer degree from National Engineers and Technicians School of Algiers (ENITA) in 1985. He received his Ph.D. Degree in Automatic Control from the Grenoble National Polytechnic Institute in 1991. In September 1991, he joined the National Engineering School of Belfort (ENIB) as Associate Professor. Since September 1999, Noureddine Zerhouni is Professor at Ecole Nationale Supérieure de Mécanique et des Microtechniques (ENSMM) in Besançon. His main research activities are concerned with intelligent maintenance systems and e-maintenance.



Fares Ben Youssef is a Reliability and COSD Manager at Schlumberger in Youngsville, United States. He has a Master degree in electrical engineering from the engineering school of Paris XI University. His interests include electrical board and component failure analysis, equipment efficiency, condition based maintenance, and PHM.



Nannan Shen is a reliability engineer at Schlumberger technology center in Clamart, France. She obtained the B.S. degree in Biochemistry in 2007 and the M.S. degree in Biochemistry in 2009 both from Shanghai Jiaotong University, China. She has accumulated a wealth of experiences in equipment operation and maintenance through field work. Her research interests include electrical board and component failure analysis, and PHM.

An End-to-End Pipeline for Uncertainty Quantification and Remaining Useful Life Estimation: An Application on Aircraft Engines

Marios Kefalas¹, Bas van Stein², Mitra Baratchi³, Asteris Apostolidis⁴, and Thomas Bäck⁵

^{1,2,3,5} *LIACS, Leiden University, Leiden, 2333 CA, The Netherlands*
{m.kefalas, b.van.stein, m.baratchi, t.h.w.baeck}@liacs.leidenuniv.nl

⁴ *Faculty of Technology, Amsterdam University of Applied Science, Amsterdam, 1097 DZ, The Netherlands*
a.apostolidis@hva.com

ABSTRACT

Estimating the remaining useful life (RUL) of an asset lies at the heart of prognostics and health management (PHM) of many operations-critical industries such as aviation. Modern methods of RUL estimation adopt techniques from deep learning (DL). However, most of these contemporary techniques deliver only single-point estimates for the RUL without reporting on the confidence of the prediction. This practice usually provides overly confident predictions that can have severe consequences in operational disruptions or even safety. To address this issue, we propose a technique for uncertainty quantification (UQ) based on Bayesian deep learning (BDL). The hyperparameters of the framework are tuned using a novel bi-objective Bayesian optimization method with objectives the predictive performance and predictive uncertainty. The method also integrates the data pre-processing steps into the hyperparameter optimization (HPO) stage, models the RUL as a Weibull distribution, and returns the survival curves of the monitored assets to allow informed decision-making. We validate this method on the widely used C-MAPSS dataset against a single-objective HPO baseline that aggregates the two objectives through the harmonic mean (HM). We demonstrate the existence of trade-offs between the predictive performance and the predictive uncertainty and observe that the bi-objective HPO returns a larger number of hyperparameter configurations compared to the single-objective baseline. Furthermore, we see that with the proposed approach, it is possible to configure models for RUL estimation that exhibit better or comparable performance to the single-objective baseline when validated on the test sets.

Marios Kefalas et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

Prognostics and health management (PHM) is a research area with multiple methodologies and functions as a *decision support tool* that aims at minimizing maintenance costs and predicting when a failure could occur by the assessment, prognosis, diagnosis, and health management of engineered systems (Nguyen et al., 2019). The core of PHM is failure prognostics. Failure prognostics refers specifically to the phase involved with predicting future behavior and the system's useful lifetime left in terms of current operating state and the scheduling of required maintenance actions to maintain system health (Vachtsevanos, Lewis, Roemer, Hess, & Wu, 2006). This useful lifetime left is often called the remaining useful life (RUL) (Nguyen et al., 2019) and is defined as the length from the current time and operating state to the end of the useful life (Si, Wang, Hu, & Zhou, 2011). The notice of pending equipment failure allows for sufficient lead-time so that necessary decisions, personnel, equipment, and spare parts can be organized and deployed, thus minimizing equipment downtime and repair costs. By leveraging RUL estimation¹, industries, such as aerospace, maritime, and energy, can improve maintenance schedules to avoid catastrophic failures and consequently save lives and costs (Zhang, Lim, Qin, & Tan, 2017). The industry has to also assure that its asset utilization is optimum by guaranteeing a timely - but not premature - maintenance. Furthermore, this practice promotes sustainability as the use of spare parts is optimum and no useful life is wasted.

The estimation of the RUL can be done in various ways. *Model-based*, *data-driven* and *hybrid* methods are the most prominent approaches (Nguyen et al., 2019), and in general all methods make some use of the sensor data of the equipment and/or maintenance history. Model-based methods (or

¹In this work we will be using the terms *RUL prediction* and *RUL estimation* interchangeably, unless otherwise stated.

physics-based methods) rely on an established mathematical model of the system in question and, as a result, call for a thorough understanding of the system's physics and processes. This can be prohibitively costly in terms of time and money due to the amount of time and domain expertise needed to develop and fine-tune such models². On the other hand, data-driven methods are relatively easier to develop as they do not need (a lot of) expert knowledge to develop the model, rendering them domain-agnostic or easily transferable between domains. They can require, however, large amounts of data. Lastly, hybrid (or fusion) methods leverage the advantages of the two previous methods while minimizing their limitations. The previous groups of methods showcase that data-driven approaches are, in general, available to a broader audience due to their domain-agnostic nature, allowing universal applicability, and also because of the plethora of tools that are developed.

Data-driven approaches either fall under the category of classic machine learning (ML) algorithms (such as random forests (RF)) (Zhang et al., 2017; Sateesh Babu, Zhao, & Li, 2016a) or the more recently proposed deep neural networks (DNNs) (Hsu & Jiang, 2018; Listou Ellefsen, Bjørlykhaug, Æsøy, Ushakov, & Zhang, 2019; Zheng, Ristovski, Farahat, & Gupta, 2017). In both cases, though, the estimation of the RUL is a challenging problem. The remaining useful life is not merely a target variable that can be predicted from sensor measurements, but it is a variable that needs to be inferred from a longer trend of degradation patterns and when those begin to occur. In this view, and due to the advances in the general field of artificial intelligence (AI), deep learning (DL) and DNNs have proven to be a successful candidate to the RUL estimation task (Lei et al., 2018; Benker, Furtner, Semm, & Zaeh, 2021; Kefalas, Baratchi, Apostolidis, van den Herik, & Bäck, 2021; Caceres, Gonzalez, Zhou, & Droguett, 2021; Peng, Ye, & Chen, 2020; B. Wang, Lei, Yan, Li, & Guo, 2020). One significant advantage of DNNs lies in their ability to learn features from raw data automatically and extract patterns that can enhance the RUL estimation accuracy (Benker et al., 2021; B. Wang et al., 2020). DNNs owe their success to their representational power and their capacity to learn sets of hierarchical features from simpler features due to their deep, multilayer architectures (Goodfellow, Yoshua Bengio, & Aaron Courville, 2016). However, most of the state-of-the-art DL approaches used in prognostics provide mainly point estimates to their RUL predictions (Peng et al., 2020; Caceres et al., 2021; Biggio, Wieland, Chao, Kastanis, & Fink, 2021). This is because DNNs do not inherently quantify the uncertainty associated with their predictions but instead treat their weights and biases as deterministic values. These predictions, though, are uncertain since they are prone to noise and wrong model inference (see Section 4.4). Specifically, there are two sources of uncertainty, namely *epistemic* (or model) uncertainty and *aleatory*

(or data) uncertainty (Hüllermeier & Waegeman, 2021). The former occurs due to inadequate knowledge, data, and representational capacity of the model and the latter due to the inherent uncertainty of the data distribution (Caceres et al., 2021; Abdar et al., 2021). Additionally, from the nature of epistemic uncertainty we can see that it is a *reducible* part of the (total) uncertainty of a modeling process, as it can be reduced on the basis of additional information. On the contrary, aleatory uncertainty is an *irreducible* part of the (total) uncertainty, due to the inherently random effects in the data-generating process (Hüllermeier & Waegeman, 2021). Most problems in engineering involve both sources of uncertainties. However, it may be difficult to distinguish whether a particular uncertainty should be put in the aleatory category or the epistemic category, in the modeling phase (Kiureghian & Ditlevsen, 2009).

The lack of a measure of uncertainty, however, can lead to overly confident decisions (Caceres et al., 2021; Gal & Ghahramani, 2016). When it comes, for example, to cost-critical or safety-critical applications, it is necessary to know how much confidence a DL method has on its prognostic results and even more so when it comes to the RUL estimation (Peng et al., 2020; Biggio et al., 2021; Benker et al., 2021; Caceres et al., 2021). In addition, even though DNNs output predictive probabilities (e.g., image classification), these probabilities are falsely interpreted as model confidence (Gal & Ghahramani, 2016). For example, the probability of the softmax on the final layer of a neural network (NN). will not reflect if the network has knowledge of the input (see also adversarial examples (Szegedy et al., 2014)). Additionally, decision-making based on a single-point estimate is error-prone and leaves no room for the decision-maker to make an actionable choice (Peng et al., 2020). When such an uncertainty estimate is available (see also Section 2) it is often the case that end-users and decision-makers need to choose by lacking broader information, such as distribution of predictions or other statistics that can assist the logistics further.

Furthermore, the end-user or researcher is faced with a multitude of decisions around the hyperparameters of the pre-processing of the data (e.g., label construction for RUL data) and of the learning algorithm (e.g., the number of layers in a DNN). Hyperparameters are not learnt but have to be set a-priori, and they have a large impact on the predictive performance of a method but also uncertainty. On top of that, there can be hyperparameter configurations that allow low prediction error but have (relatively) large uncertainty and vice versa. In such scenarios, where trade-offs exist, it is vital to move towards a more *user-centric* approach, where the end-user can decide which hyperparameter configuration to adopt based on the criticality of the task. As such, hyperparameters need to be considered carefully both in terms of model accuracy and uncertainty estimates.

²Model-based methods do not require (a lot of) historical data for their development, making them the only option for the development of models for new systems.

The aforementioned statements motivate our main research question: Can we propose an automated framework for configuring RUL prediction models which are highly accurate and have less estimation uncertainty?

More specifically, our contributions are as follows:

1. We automatically optimize the hyperparameters of the Bayesian deep learning (BDL) model through Bayesian multi-objective optimization, jointly minimizing the RUL prediction error and the combined aleatory and epistemic uncertainties of the estimations. The reasoning behind this is that in certain tasks, there can be conflicts between these two objectives, as we briefly mentioned previously.
2. Together with the model hyperparameters, we further optimize the hyperparameters which are specific to the task of RUL estimation (the RUL label construction, see also Section 4), which is known to have an effect on the algorithmic performance (Sateesh Babu, Zhao, & Li, 2016b). We provide thus, a thorough, end-to-end approach that can further assist researchers and end-users for offline RUL estimation.
3. We adopt a *user-centric* approach that allows the user to estimate the RUL based on the model output, as it promotes a more interpretable RUL decision. We demonstrate how survival curves can provide the end-user with information regarding the RUL and its confidence.
4. We evaluate our multi-objective hyperparameter optimization (HPO) approach against a single objective HPO by taking the harmonic mean (HM) of the objectives. Our approach is validated on two subsets of the widely used C-MAPSS dataset (A. Saxena & K. Goebel, 2008).

The rest of the paper is organized as follows. In Section 2, we present related work in this field and in Section 3, we formally define the problem of the RUL estimation. In Section 4, the proposed method and its modules are introduced and in Section 5 we present the dataset used and discuss the experimental results. Finally, in Section 6 we conclude and discuss the limitations of our framework and suggest future work.

2. RELATED WORK

The field of PHM has been widely credited in the past years with numerous contributions from researchers. Academic interest, industrial applications, as well as the scientific challenge of developing methods to forecast a failure, have been the driving forces. While model-based prognostic methods, such as Kalman filters and their variants (Govaers, 2019; Kalman, 1960), take into account the modeling and data uncertainty, only a few studies in the data-driven domain address this matter, despite its importance (Biggio et al., 2021). Touching upon the previous statement, in this section, we will present related work in the context of uncertainty quantification (UQ) for the RUL estimation, attending only to data-driven approaches.

From the traditional ML methods, only Gaussian process regression (GPR) (Rasmussen & Williams, 2006) (also known as Kriging) addresses UQ. GPR is a stochastic interpolation method where unseen locations of a stochastic process are estimated as a linear function of observed values. It can further be understood as a form of Bayesian Inference (BI). Specifically, GPR places a Gaussian prior over the functions that could have generated the observed data. Using Bayes's theorem by combining the Gaussian prior and the Gaussian likelihood function (for tractability), we get the predictive distribution for a new value. However, GPR might not be the optimal model for some data, e.g., the data does not come from a Gaussian process, or the dimensionality is high. Furthermore, the data generating the predictions are not learned automatically as in DL but need proper pre-processing (e.g., feature extraction), and another downside of GPR is the GPR variance, of which is known that it can be over-optimistic (den Hertog, Kleijnen, & Siem, 2006).

In this view, from the data-driven approaches, we will only review recent work that adopted a DL solution. We made this decision because, as also mentioned in Section 1, DL is becoming prominent in data-driven prognostics, as well as there has recently been a lot of attention on UQ for DL (Gal & Ghahramani, 2016; Blundell, Cornebise, Kavukcuoglu, & Wierstra, 2015; Osband, Aslanides, & Cassirer, 2018; Abdar et al., 2021). This collection is by no means exhaustive. We refer the interested reader to (Nguyen et al., 2019) and (Krishna & Baghaei, 2019) for a more thorough overview of related work on PHM.

Epistemic Uncertainty The work by Peng et al. (Peng et al., 2020) is a recent data-driven example of UQ in prognostics. The authors present a DL approach from a Bayesian viewpoint to address the confidence of their RUL predictions and implement the Bayesian approximation using Monte Carlo Dropout (MC Dropout) (Gal & Ghahramani, 2016) (see also Section 4.4). Kraus et al. (Kraus & Feuerriegel, 2019) dealt with epistemic uncertainty in prognostics using variational inference (VI) (see also Section 4.4) and combine DL with notions from survival analysis to increase the *intepretability* of the estimation. In the same domain, Wang et al. (B. Wang et al., 2020) used MC Dropout to estimate the epistemic uncertainty of a recurrent convolutional neural network (RCNN) for the RUL estimation. However, none of the previous studies touched upon aleatory uncertainty.

Aleatory Uncertainty Zhao et al. (Zhao, Wu, Wong, Sun, & Yan, 2020), addressed the aleatory uncertainty by using a deep convolutional neural network (DCNN) through a shortened version of the ResNet (He, Zhang, Ren, & Sun, 2015) and assumed that the target RUL values follow a Gaussian distribution with parameters μ and σ being the network's outputs. They also adopted a non-parametric approach by combining the predicted RUL from the network with quantile regression, predicting this way multiple RUL at different quantile levels.

However, this approach did not take into account epistemic uncertainty and, to the extent of our knowledge, there was no HPO.

Epistemic and Aleatory Uncertainties Caceres et al. considered in (Caceres et al., 2021) both epistemic and aleatory uncertainties. They used an explicit form of VI to account for the epistemic uncertainty and addressed the aleatory uncertainty by a probabilistic output layer parameterized by a Gaussian distribution and further performed HPO through grid search. In the same manner, Kim et al. (Kim & Liu, 2021) and Li et al. (G. Li, Yang, Lee, Wang, & Rong, 2021) designed RUL frameworks by taking into account the effects of both epistemic and aleatory uncertainties. They both used MC dropout to address the epistemic uncertainties. Kim et al. (Kim & Liu, 2021) addressed the aleatory uncertainty by a probabilistic output layer parameterized by a Gaussian distribution and assumed a monotonically decreasing relationship between the aleatory uncertainty and RUL and further performed HPO on the number of hidden layers amongst other hyperparameters. Li et al. (G. Li et al., 2021) modeled aleatory uncertainty by a probabilistic output layer following various types of life-time distributions (Weibull, Gaussian, and Logistic). Benker et al. (Benker et al., 2021) adopted a Bayesian neural network and addressed both uncertainties as well, but took into account the aleatory uncertainty post-training. They further quantified the epistemic uncertainty using a Hamiltonian Monte Carlo method, a more efficient variant of the Markov Chain Monte Carlo (MCMC) methods in high dimensional spaces.

These recent studies have made a great contribution to the field of data-driven prognostics by proposing methods to account for and quantify the uncertainty of their predictions. Nonetheless, there remain perspectives to consider. In more detail, most of the literature reviewed ((Zhao et al., 2020; G. Li et al., 2021; Benker et al., 2021)) did not state any form of HPO and those that did ((Peng et al., 2020; Caceres et al., 2021; Kim & Liu, 2021; Biggio et al., 2021)), did not optimize necessary hyperparameters in the pre-processing stage and used less efficient HPO techniques (e.g., grid search). What is more, the reviewed methods that perform some form of HPO used *only* the RUL prediction error as the only criterion to guide the HPO, as opposed to also taking into account the epistemic and aleatory uncertainties. Lastly, in our literature review, we did not come across any methods that allow the end-user to make an informed RUL prediction based on information output by the model.

3. PROBLEM DEFINITION

The RUL of an asset or system is defined as the length from the current time and operating state to the end of the useful life (Si et al., 2011). Because the adjective *useful* is subjective, the previous definition can be extended to the time when the extent of deviation or degradation of the performance from its expected normal operating conditions exceeds a *pre-defined*

threshold (Saxena, Goebel, Simon, & Eklund, 2008), when the system needs to be repaired or retracted. Based on this, we can define the RUL at time $t \in \mathbb{R}_{\geq 0}$ as:

$$RUL(t, \mathbf{D}_t) = \inf\{s \in \mathbb{R}_{\geq 0} : s \geq t \wedge \mathbb{1}_{\mathbb{S}^c}(CI(s, \mathbf{D}_t))\} - t, \quad (1)$$

where \inf represents the infimum of a set and $\mathbb{1}$ is the indicator function. \mathbb{S} is a user-defined system operating envelope. The operating envelope is a collection of boundary limits that put the integrity of an asset at risk when exceeded. CI represents a user-specified condition index, which monitors if the asset has exceeded its operating constraints. In this case, the CI lies in the complement of \mathbb{S} (\mathbb{S}^c), which indicates that the system must be repaired or maintained.

The time t denotes the time at which the prediction needs to be performed. \mathbf{D}_t represents the data generated by an asset used for the RUL prediction of that asset. Most commonly \mathbf{D}_t is sensor measurements recorded in time (e.g., pressure, temperature) accompanied by event labels (e.g., times-to-failure), up until time t . In principle, though, \mathbf{D}_t can be any type of data, structured or not, that can facilitate the estimation.

The quantity $\inf\{s \in \mathbb{R}_{\geq 0} : s \geq t \wedge \mathbb{1}_{\mathbb{S}^c}(CI(s, \mathbf{D}_t))\}$ in Equation 1 can also be referred to as the *end-of-life* (EoL), to mark that the system’s “life”, based on user-defined criteria, has come to an end. Ultimately the estimation of RUL amounts to the approximation of the EoL. We should note that the EoL does not necessarily mean that the system has gone through a catastrophic failure but might operate sub-optimally according to user-defined criteria.

Finally, from a *data-driven* perspective, the estimation of the RUL of an asset involves creating a model which is trained on data from the same type of assets. Let U be the set of training data. Each instance $u \in U$ is presented as a multivariate time-series of sensor readings $\mathbf{X}_u = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{T(u)}]^T \in \mathbb{R}^{m \times T(u)}$, with $T(u)$ time-steps where the last time-step corresponds to the end-of-life (EoL) of the unit u . Each point $\mathbf{x}_t \in \mathbb{R}^m$ is an m -dimensional vector corresponding to readings from m sensors at time t .

4. PROPOSED METHOD

Our method works by training a Bayesian deep learning model on training data U presented in the form of multivariate time-series. The steps of our method are summarized as:

1. Data pre-processing by removing any redundant signals, normalizing the remaining sensor values and performing a sliding window transformation.
2. Target-RUL construction to allow supervised learning.
3. Modeling using a BDL model and taking into account the uncertainty of the predictions.
4. Hyperparameter optimization of the hyperparameters of steps 1,2, and 3.

4.1. Pre-Processing

Sensor selection is an initial step of pre-processing multivariate time-series data. It involves filtering the available data from sensor measurements which, for example, either do not exhibit any correlation with the target or have strong correlations with other sensors. In the latter case, we usually discard some of the correlated features. Furthermore, even if no correlation is present, but the sensors do not exhibit any variation, these features can often be discarded as they do not add any valuable information. What is more, having a large number of sensors is not always beneficial for training models as it increases the chance of overfitting.

Pre-processing also involves normalizing the available data to mitigate any effect that different ranges of values or large deviations can have in the subsequent learning phase. Two of the most often used normalization methods are Z-normalization and Min-max normalization:

- Z-normalization (or standardization): This normalization transforms the data into having 0 mean and unit variance as: $x' = (x - \mu) / \sigma$;
- Min-max normalization (or rescaling): This normalization maps the range of the data into $[0, 1]$ or more generally into $[a, b]$ as: $x' = a + \frac{(x - \min(S))(b - a)}{\max(S) - \min(S)}$,

where S is a feature (e.g., a sensor), x, x' are the value and the transformed value of the feature S , and μ, σ are the mean and standard deviation of S , respectively. In addition, a, b are the lower and upper bounds of the projection, and $\min(S), \max(S)$, are the minimum value and maximum value of S , respectively. Normalization is applied on every sensor/feature independently.

As a next step, for each X_u , we perform a sliding window transformation with a sequence of length w (window size), in order to enclose the inputs into multidimensional sequential data, which are to be considered as one sample. This transformation allows one to increase the number of training data, standardize the sample input lengths, and accelerate model training (Caceres et al., 2021). *For this work, the window size w is treated as a pre-processing hyperparameter.*

4.2. Target-RUL Construction

We would like to tackle this problem as a regression problem. However, one of the main challenges of RUL estimation is the lack of ground-truth values (Sateesh Babu et al., 2016b). In most cases, the only available data are the data from the sensor measurements. However, these data are not labeled with any information regarding the RUL, such as the times-to-failure. The latter is essential for the training procedure as it carries decisive information that will allow the learner to uncover rules that estimate the RUL given sensor measurements. There are two popular ways to create these labels, namely *linear* and *piece-wise linear* methods (Sateesh Babu et al., 2016b).

The former interprets the RUL in the strictest sense, as time to failure. Thus, every time-step is mapped to a value equal to $EoL - t$, where t is the current time-step. This approach, however, implies that the health of the system degrades linearly with usage (Sateesh Babu et al., 2016b). The latter reflects the fact that initially the degradation is negligible, and after a specific point in time, it becomes more evident (see Figure 1 for an example). The point after which the RUL degrades linearly is called the *reflection point* (Hsu & Jiang, 2018). This, way we can construct an RUL curve for each $u \in U$, by mapping each rolling window to the RUL *at the end of that window*. *For this work, the type of label creation method and the reflection point are treated as pre-processing hyperparameters.*

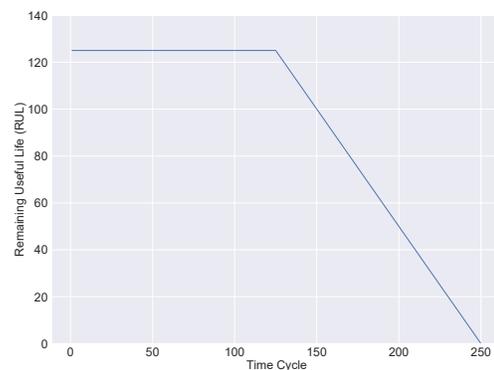


Figure 1. Toy example of a piece-wise linear RUL target function. The reflection point is at time cycle 125. Adapted from (Kefalas et al., 2021).

4.3. Modeling

As mentioned in Section 1, amongst the data-driven methods employed for prognostics DNNs have proven to be good candidates due to their representational power (Lei et al., 2018; Benker et al., 2021; Kefalas et al., 2021; Caceres et al., 2021; Peng et al., 2020; B. Wang et al., 2020). In general, shallow learning methods are not designed for large-scale datasets and, more importantly, need extensive feature engineering efforts (Zhou, Droguett, Mosleh, & Chan, 2021). In this view, we decided to employ DL to address the RUL estimation problem. As this task is based on sequential data (multivariate time-series), we decided to use recurrent layers and specifically gated recurrent unit (GRU) layers as the model base due to their lower complexity and similarly good performance in modeling long dependencies (G. Li et al., 2021), when compared to long short-term memory (LSTM) layers.

4.4. Uncertainty Quantification

As discussed briefly in Section 1 predictions made by neural networks are inherently uncertain, as they are prone to noise and/or wrong model inference. At the same time, however,

NNs treat their weights and biases as deterministic values. This results in NNs being overly confident, even when they should *not* be. In general, there are two sources of uncertainty. In the context of NN, the epistemic and aleatory uncertainties can be considered by putting a prior on model parameters or the outputs. The latter means assuming that the model outputs follow a specific distribution, such as Weibull. The former can be addressed by treating the weights and biases (we jointly note them as W) of the network as random variables, defining a prior over them, and then using Bayesian inference to learn the posterior distributions of the network's weights (Peng et al., 2020; Zhou et al., 2021; Caceres et al., 2021) as:

$$p(W|X, Y) = \frac{p(Y|X, W)p(W)}{p(X, Y)}, \quad (2)$$

where X, Y are the training data and their labels, respectively. The posterior distribution on the network's parameters is, however, computationally intractable even for NNs of any practical size, as the number of parameters is very large and the functional form of a NN does not allow for exact integration (Blundell et al., 2015; Gal & Ghahramani, 2016; Caceres et al., 2021). Moreover, the denominator in Equation 2 is unavailable in closed form or requires exponential time to compute (Blei, Kucukelbir, & McAuliffe, 2017).

A large part of ongoing research is focused on approximating such posterior distributions (Biggio et al., 2021). Amongst these, prominent methods are Markov Chain Monte Carlo (MCMC) methods and its variants and variational inference (VI) (Biggio et al., 2021; Zhou et al., 2021; Blei et al., 2017; Caceres et al., 2021). The former, generally, converge slowly and are computationally expensive for large datasets or complex models. Instead, VI solves the same problem by using optimization techniques rather than sampling methods like MCMC (Blei et al., 2017). Specifically, VI sidesteps the difficulty mentioned above altogether by defining an approximate *variational* distribution $q(W)$ from a distributional family \mathbb{D} , that is the best approximation to the exact posterior $p(W|X, Y)$, with respect to the Kullback-Leibler (KL) divergence. This means that,

$$q(W) = \operatorname{argmin}_{q(W) \in \mathbb{D}} KL(q(W)||p(W|X, Y)), \quad (3)$$

where $KL(q(W)||p(W|X, Y))$ is defined as:

$$KL(q(W)||p(W|X, Y)) = \mathbb{E}_{q(W)} \left[\log \frac{q(W)}{p(W|X, Y)} \right] \quad (4)$$

However, because Equation 3 is intractable³ VI maximizes instead what is called the evidence lower bound (ELBO), which is defined as:

$$ELBO(q) = \mathbb{E} [\log(p(X, Y|W))] - KL(q(W)||p(W)) \quad (5)$$

³See (Blei et al., 2017) page 6 for details.

In turn, though, exactly maximizing Equation 5 is computationally prohibitive. To address this, VI can be divided into methods that implicitly use model uncertainties, such as MC Dropout (Gal & Ghahramani, 2016) and methods that explicitly model weight parameters as probability distributions such as Bayes-by-Backprop (Blundell et al., 2015; Caceres et al., 2021; Zhou et al., 2021).

In this work, we have decided to use MC Dropout to model the **epistemic uncertainty** due to its simplicity, scalability, and computational efficiency compared to other Bayesian deep learning approaches (Gal & Ghahramani, 2016; Kim & Liu, 2021). It is implemented through gradient-based learning methods and stochastic regularization techniques, which are widely available in existing DL libraries (Peng et al., 2020). MC Dropout is, in essence, regular dropout applied at both training and inference steps. The addition of dropout between every layer can switch off some portion of neurons in each layer and generate random predictions as samples from a probability distribution that is considered equivalent to performing approximate VI. In more detail, MC Dropout showed that by choosing a specific form of an approximate distribution q , as a distribution over matrices whose columns are randomly set to zero, the VI in a NN can be interpreted as performing one forward pass through the NN with dropout. For more details on MC Dropout, see (Gal & Ghahramani, 2016) and the accompanying appendix.

We should note here that there is a current debate as to the validity of MC Dropout being Bayesian (Caceres et al., 2021; Zhou et al., 2021; Osband et al., 2018). In (Osband et al., 2018), Osband et al. in highlighted that a shortcoming of MC Dropout is that the dropout rate does not depend on the data, which translates into the fact that employing dropout for posterior approximation cannot say anything about a set of data being observed once or more times. This, of course, can have significant implications in support of reliable uncertainty quantification and consequently deserves attention. As this work was mainly devoted to the usage of bi-objective HPO and user-centric approach, we have decided to address this *highly relevant but challenging issue* in future work.

Finally, in order to model the **aleatory uncertainty**, inspired by (Martinsson, 2016), we further assume that the RUL values follow a Weibull distribution, the reason being that Weibull is extensively employed in survival and reliability analysis to model times-to-failure. Moreover, it is simple, but also expressive, being able to take various forms, such as the exponential distribution (G. Li et al., 2021). The probability density function (PDF) of the 2-parameter Weibull that we used is defined as: $f(x) = \frac{\beta}{\alpha} (\frac{x}{\alpha})^{\beta-1} e^{-(x/\alpha)^\beta}$, for $x \geq 0, \alpha, \beta \in (0, +\infty)$, where α is the scale parameter and β the shape parameter of the distribution.

In this view and to adopt a user-centric approach for the RUL estimation (3rd contribution), the output layer of the DNN

(see Section 4.3) will output the parameters of the Weibull distribution, α, β . This is a more *user-centric* approach, as for a sample input (e.g., a sequence of sensor values), the end-user is presented with the parameters that govern the distribution of the times-to-failure. This allows for more informative and interpretable decision-making in subsequent steps. The end-user can decide himself what statistics or percentiles (e.g., the mean-time-to-failure (MTTF)) to use as the point estimate of the RUL and the overall knowledge of the distribution of failure times can allow decision-makers to reason if the results are plausible or not. This contrasts with most methods that return a point-estimate to the end-user.

4.5. Hyperparameter Optimization

The optimization of hyperparameters enhances the performance of a machine learning algorithm, and thus, HPO is considered an important step in developing AI and ML frameworks.

Various methods and algorithms are available for HPO, such as grid search (GS), random search (RS), evolutionary algorithms (EA), and Bayesian optimization (BO) (Feurer & Hutter, 2019). In this study, a bi-objective variant of a state-of-the-art BO algorithm, namely *Mixed-integer Parallel Efficient Global Optimization* (MIP-EGO), is chosen due to its efficiency for optimizing expensive problems (Stein, Wang, & Back, 2019). MIP-EGO is based on Efficient Global Optimization, also known as Bayesian Optimization (BO). The algorithm uses random forests (RFs) models to handle mixed integer data and mixed integer evolution strategies (MIES) as internal optimizer. The bi-objective variant of MIP-EGO uses the S-metric hyper-volume (see also Section 5.3) improvement infill criterion to select new candidate solutions.

In order to perform the HPO of the Bayesian deep learning and the problem-specific pre-processing hyperparameters by jointly optimizing the prediction error and uncertainty address (1st and 2nd contributions), MIP-EGO is set to determine the hyperparameter values that *minimize simultaneously* the point-wise root mean squared error (RMSE) and the uncertainty by optimizing the bi-objective function described in Algorithm 1. In more detail, MIP-EGO will evaluate different configurations h_p by preprocessing the data and training a DNN (lines 1 and 2). In lines 3 – 19 the trained network is used to make predictions on each sample of the validation set (size m) by multiple passes R which output different α, β at each pass using MC Dropout (see Section 4.4). To determine the RUL estimate for an input sample, we calculated the median of the predicted α s (\bar{a}) and the median of the predicted β s (\bar{b}) (line 11) and used the mean-time-to-failure (MTTF) of the Weibull distribution with parameters the calculated medians (line 15). The choice of the MTTF was to reduce the selection bias to any statistic and the choice of median to counteract effects of possible outliers. Of course, any

other statistic could be used here. The mean-time-to-failure is defined as: $MTTF(\alpha, \beta) = \alpha\Gamma(1 + 1/\beta)$, where Γ is the gamma function. For the over all point-wise performance, f_1 , we calculated the RMSE between the predicted RUL (over all the instances) and the ground truth values (line 18). To determine the uncertainty for an input sample, we calculated the standard deviation of the predicted α s (\hat{a}) and the standard deviation of the predicted β s (\hat{b}) (line 13) and averaged the two values. For the overall uncertainty f_2 , we calculated the average over all the uncertainties (line 19).

Algorithm 1: Bi-objective Function

```

Data:  $X, V, hp, R$  # Training data, validation data,
hyperparameter configuration, sample size
Result:  $f_1, f_2$  # RMSE, uncertainty
1  $X', V', Y_{X'}, Y_{V'} \leftarrow \text{Pre\_processing}(X, V, hp);$  # Data
pre-processing and RUL creation for the training and
validation data (see Sections 4.1 and 4.2)
2  $M \leftarrow \text{DNN}(X', V', Y_{X'}, Y_{V'}, hp);$  # Model training
3  $m \leftarrow |V'|; RUL \leftarrow \emptyset; Var \leftarrow \emptyset;$ 
4 for  $i \leftarrow 1$  to  $m$  do
5 |  $A \leftarrow \emptyset; B \leftarrow \emptyset;$ 
6 | for  $j \leftarrow 1$  to  $R$  do
7 | |  $a, b \leftarrow M(V_i);$ 
8 | | # Predicting using trained DNN through MC
Dropout (see Section 4.4)
9 | |  $A \leftarrow A \cup a; B \leftarrow B \cup b;$ 
10 | end
11 |  $\bar{a} \leftarrow \text{median}(A); \bar{b} \leftarrow \text{median}(B);$ 
12 | # Median values of A and B
13 |  $\hat{a} \leftarrow \text{std}(A); \hat{b} \leftarrow \text{std}(B);$ 
14 | # Standard deviations of A and B
15 |  $RUL \leftarrow RUL \cup \mathbb{E}[\text{Weibull}(\bar{a}, \bar{b})];$ 
16 |  $Var \leftarrow Var \cup \text{mean}([\hat{a}, \hat{b}]);$  # average between  $\hat{a}, \hat{b}$ 
17 end
18  $f_1 \leftarrow \text{RMSE}(RUL, Y_{V'});$ 
19  $f_2 \leftarrow \text{mean}(Var);$  # Average value of Var
20 Return  $f_1, f_2$ 

```

5. EXPERIMENTAL SETUP AND RESULTS

We are interested in investigating the existence and trade-offs between the RUL prediction error and the prediction uncertainty when using bi-objective HPO, and to examine the advantages that can be gained compared to using a single-objective variant. Furthermore, we show how the proposed method can be more user-centric compared to the current techniques. Datasets and experimental results are described in this section.

5.1. Data

In this study, we use the widely used C-MAPSS benchmark dataset (A. Saxena & K. Goebel, 2008). The dataset was released in 2008 (Saxena et al., 2008) and it has been used in the field of PHM ever since, to develop techniques and methods for estimating the RUL (Ramasso & Saxena, 2014; Krishna &

Baghaei, 2019). It is a simulated turbofan engine degradation dataset from NASA’s Prognostics Centre of Excellence⁴. The dataset consists of four subsets: FD001, FD002, FD003, and FD004, each of which exhibits a different number of operating conditions and fault modes. In this work, we used datasets FD001 and FD003, which exhibited the same number of operating conditions but different number of fault modes. Each of these datasets is arranged in an $n \times 26$ matrix where n corresponds to the number of data points (samples) in each unit and 26 is the number of columns/features. Each row is a snapshot of data taken during a single operating time cycle. Regarding the 26 features, the 1st represents the engine number, the 2nd represents the operational cycle number. Features 3 – 5 represent the operational settings, and columns 6 – 26 represent the 21 sensor values. Engine performance can be significantly affected by the three operating settings. More information about these 21 sensors can be found in (Ordóñez, Sánchez Lasheras, Roca-Pardiñas, & Juez, 2019). What is more, each subset exhibits a different number of faults (see Table 1).

Each of these subsets are further split into training set and test set (see Table 1 for details). For each engine trajectory within the training sets, the last data entry corresponds to the end-of-life (EoL) of the engine, i.e., the moment the engine is declared unhealthy or in failure status. The test sets contain data up to some time before the failure and the aim here is to predict the RUL for each of the test engines.

These multivariate time-series are from a different engine i.e., the data can be considered to be from a fleet of engines, of the same type though, and each trajectory is assumed to be the life-cycle of an engine. Every engine starts with different degrees of initial wear and manufacturing variation which is unknown to the user. This wear and variation is considered normal, i.e., it is not considered a fault condition.

To compare the model performance on the test data, we need some objective performance measures. In this study, we use the *Root Mean Square Error (RMSE)* (Zheng et al., 2017; Listou Ellefsen et al., 2019; X. Li, Ding, & Sun, 2018), defined as: $RMSE = \sqrt{1/n \sum_{i=1}^n d_i^2}$, where $d_i = R\hat{U}L_i - RUL_i$, $R\hat{U}L_i$ is the estimated RUL and RUL_i is the ground truth RUL for instance (engine) i , respectively.

5.2. Experimental Setup

The experiments⁵ were executed on 10 *NVIDIA Tesla T4* GPUs, of 16GB, *GDDR6* memory. Source code has been developed in *Python V3.8.8*⁶. Experimental time was around 3-5 days (wall clock time), per dataset.

⁴<https://ti.arc.nasa.gov/tech/dash/groups/pcoe/>

⁵The source code of the experiments can be found at <https://github.com/MariosKef/RULe>.

⁶We used *tensorflow(2.5.0)*, *scikit-learn(0.24.1)*, *pandas(1.2.3)*, *numpy(1.19.5)*.

Table 1. FD001 and FD003 C-MAPSS dataset details

Data-Set	FD001	FD003
Train trajectories	100	100
Test trajectories	100	100
Operating conditions	1	1
Fault conditions	1	2
Max train trajectory (cycles)	362	525
Min train trajectory (cycles)	128	145
Max test trajectory (cycles)	303	475
Min test trajectory (cycles)	31	38
Training samples	20631	24720

We began by randomly selecting 80% of units from the training set and using the remaining 20% as the validation set to select the hyperparameters. We then randomly truncate the trajectories of the validation set at five different locations such that five different cases are obtained from each trajectory following (Malhotra et al., 2016). The truncation is needed to replicate the dedicated test data, i.e., trajectories up to some time before the failure. Note here, however, that we did not use any information from the dedicated test set. Minimum truncation is 5% of the total life, and maximum truncation is 96% of the total life. We continued with the pre-processing of the training and validation sets. In more detail, we normalized the data transforming the 3 operational settings and 21 sensor values to the range $[-1, 1]$ (min-max normalization) and discarded any of them that have zero variance. Constant values do not provide any useful degradation information for determining the RUL.

For the next steps of the pre-processing and data transformation (sliding window and RUL target construction), as well as for the DNN training, we performed HPO to select their optimal hyperparameter values that optimize *simultaneously* the pointwise RMSE and the uncertainty, in order to address our 1st and 2nd contributions (see Section 4.5). The tuned hyperparameters and their respective ranges can be seen in Table 2. Note that the search space contains not only integer variables but also categorical ones. We executed the hyperparameter optimization (see Section 4.5) with a budget of 300 function evaluations (of which 100 are initial configurations sampled with the latin hypercube sampling (LHS) method). Moreover, the MIP-EGO configurator is set to evaluate 10 configurations per step in parallel for FD001 and 9 configurations for dataset FD003⁷.

Following the hyperparameter optimization phase, we are presented with a two-dimensional set of points showing the RMSE and UQ on the *validation set*. Each point corresponds to a specific hyperparameter configuration. By considering only the non-dominated solutions, we end up with (an approximation to) the Pareto front. The Pareto front is set of points, which cannot be improved with respect to one objective without making another objective worse (Emmerich & Deutz,

⁷This was a result of GPU availability. In any case, this did not affect the validity of the computations.

2018) (see blue points in Figure 2). The non-dominated set of solutions delivers hyperparameter configurations which allow us to view the trade-offs between the RMSE and the UQ. We can subsequently pre-process and train on the *entirety* of the training data (training and validation) using the configurations corresponding to the points on the Pareto front and finally test our method on the dedicated test set. During this stage, we use Algorithm 1 by inputting as X the entire training set, V the dedicated test set, and h_p the configuration corresponding to the selected point from the Pareto front.

Additionally, we used the Adam optimizer (Kingma & Ba, 2017) with a clip value of 0.5, $R = 30$ for the number of MC Dropout passes, and trained for 100 epochs with early-stopping (patience = 5). Finally, since we want our DNN to learn the relationship between the input sequences and the Weibull parameters, we used as a loss function the *negative log-likelihood of the 2-parameter Weibull distribution* (Yang, Ren, & Hu, 2019; Martinsson, 2016) to train the network.

5.2.1. Baseline

We also performed a baseline experiment to evaluate the bi-objective hyperparameter approach. Our baseline differs from the work we reviewed in Section 2, as none of the related work took into account the joint optimization of the RMSE and the uncertainty. Our baseline transforms the bi-objective optimization problem into a single-objective by minimizing the harmonic mean (HM) of the RMSE and uncertainty, as:

$$HM = \frac{2}{RMSE^{-1} + Uncertainty^{-1}} \quad (6)$$

For this task we used the single-objective MIP-EGO, which uses the so-called Moment-Generating Function (MGF) based infill-criterion (H. Wang, van Stein, Emmerich, & Back, 2017) to select new candidate solutions. Moreover, the MIP-EGO configurator is set to evaluate 10 configurations per step in parallel for FD001 and FD003, for a maximum of 300 function evaluations. We used this baseline in order to investigate the benefits of using the bi-objective HPO compared to the single-objective approach. The reason of taking the HM compared to e.g., the arithmetic mean, is because it is less susceptible to fluctuation of the observations, thus making it a more ideal baseline for this first study.

5.3. Hypervolume Indicator

To compare the bi-objective HPO approach to the single-objective approach based on the HM we decided to use the hypervolume indicator (HVI). The HVI or S-metric (Zitzler, Deb, & Thiele, 2000) is the hypervolume in the objective space \mathbb{R}^m that is dominated by the Pareto points bounded by a reference point $y_{ref} \in \mathbb{R}^m$. The reason for choosing the HVI as a measure of comparison is that it is intuitive, as dominating a large part of the objective space is desirable. Furthermore, the

HVI is widely used in evaluating the performance of various multi-objective optimization algorithms.

5.4. Results and Discussion

Having generated the Pareto front of the hyperparameter configurations (see Section 5.2) we selected each configuration, trained on the entirety of the dataset and made inferences about both the training and (dedicated) test data.

Figures 2 and 3 show in blue circles the Pareto front of the hyperparameter configurations performance on the *validation sets* of datasets FD001 and FD003, respectively. The red triangles depict the results on the dedicated test set (dominated solutions might exist). The number next to each point represents the hyperparameter configuration giving rise to that specific solution and are shown here to manifest how the solutions' topology changes when validated on the dedicated test set.

In order to see if the neural network can learn from the data, in Figures 4 and 5, we show the evolution, over time, of the Weibull PDFs, of units 2 and 9 from the FD001 and FD003 training data, respectively. We do this by plotting the Weibull PDFs per time-index of the units' data. For this task, we used the models which returned the lowest RMSE on the *dedicated test sets* of FD001 and FD003 (points with green shade in Figures 2 and 3). In the Figures, we can see that as the time-index of the data increases (darker-red shades in the legend), the PDFs variance decreases. Even though the distributions' variance does not initially seem to be monotonically decreasing, as we approach the end-of-life of the assets (darker-red shades), we can see that the variance decreases, giving more mass to the expected time-to-failure, and that the expected time-to-failure approaches 0. This is a desirable property as it indicates that the model can learn the correct *failure dynamics* because the more time-steps have passed, the more data has been collected, and consequently, there is more degradation information, especially near the end-of-life of the asset.

In Figures 6 and 7 we show the evolution of the HVI per a maximum of 300 function evaluations between the bi-objective and single-objective HPO. To be able to compare the HVI of the single-objective approach to the bi-objective approach, we calculated the HVI of the Pareto efficient solutions of the RMSE and uncertainty as pre-images of the HM. Furthermore, we normalized both objectives to $[0, 1]$ and used as $y_{ref} = (1.1, 1.1)$.

We can see from the two figures that the HVI of the single-objective approach and the bi-objective approach plateau to the same final HVI, albeit the bi-objective approach reaches the plateau in fewer iterations, on FD001, whereas on FD003, the single-objective approach reaches the plateau in slightly fewer iterations than the bi-objective method. The HVI might indicate that the harmonic mean manages to also identify a

Table 2. Hyperparameters in the model development for the C-MAPSS dataset

Type	Hyperparameter	Search Space
Pre-processing	Sliding window size	[20, 50]
	Reflection point (percentage of total life)	[25, 75]
	Initial RUL value	[110, 130]
	RUL degradation style	['linear', 'nonlinear']
DNN	Number of recurrent layers	[1, 3]
	Number of dense layers	[1, 3]
	Number of neurons per layer	[10, 100]
	Activations	['tanh', 'sigmoid']
	Recurrent dropout rate	[1e-5, 0.9]
	Dropout rate	[1e-5, 0.9]
	Output activations	['softplus', 'exp']
	Learning rate	[1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 2e-5]
	Batch size	[32,64,128]

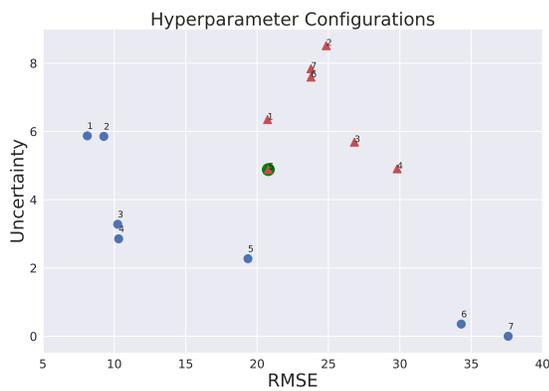


Figure 2. RMSE-UQ points corresponding to the hyperparameter configurations on FD001 using the bi-objective approach. Blue circles are the Pareto front as calculated on the validation set. The red triangles are the points calculated on the dedicated test set.

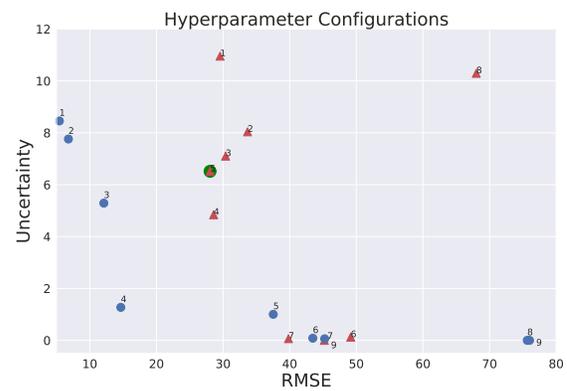


Figure 3. RMSE-UQ points corresponding to the hyperparameter configurations on FD003 using the bi-objective approach. Blue circles are the Pareto front as calculated on the validation set. The red triangles are the points calculated on the dedicated test set.

balance between the objectives and can be used as an alternative to the bi-objective HPO. The seemingly smaller number of function evaluations of the single-objective approach in the figures, compared to the bi-objective approach, is simply an artifact of infeasible configurations that were discarded by the single-objective MIP-EGO.

Examining Figures 2 and 8 we can see that the bi-objective approach returned more hyperparameter configurations lying on the Pareto front (7 blue points on Figure 2) compared to the single-objective approach (6 blue points on Figure 8). Even though the number is marginally larger, this might suggest that the bi-objective approach might be more suitable for identifying a more diverse set of hyperparameters. Moreover, it is interesting to see that the configurations returned from the two HPO methods (blue points in Figures 2 and 8) present similar values of uncertainty, even though more than 80% of the configurations of the single-objective HPO exhibit uncer-

tainty lower than 2, with that number being around 29% for the bi-objective HPO. Regarding RMSE, however, we observe the inverse trend. In the bi-objective method, more than 70% of the returned configurations result in RMSE lower than 20, with this number being 50% in the single objective approach. In addition, we can see that the performance of the resulting hyperparameters (blue points) on the dedicated test set (red triangles) differs between the two figures. Firstly, in the bi-objective approach, the performances on the dedicated test set per hyperparameter configuration are clustered together when compared to the single-objective approach in Figure 8 where the points are spread out more, especially in the uncertainty axis. Secondly, in the bi-objective method, the RMSE and uncertainty values of the dedicated test set lie in the range of [20.73, 29.82] and [4.88, 8.51], respectively. In the single-objective method these ranges are [25.97, 37.51] and [0, 7.93], respectively, for the RMSE and uncertainty. It is interesting

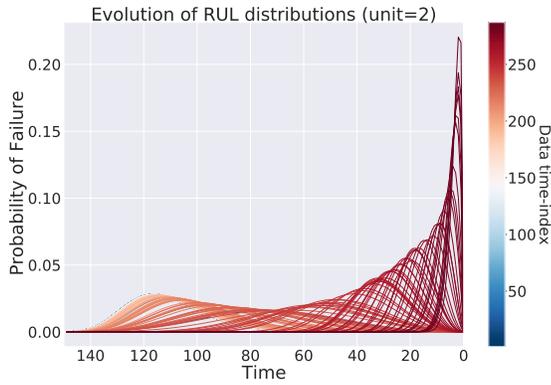


Figure 4. Evolution of Weibull distributions of unit 2 from FD001. Blue shades indicate the start of the unit’s trajectory and red shades the end. Note that the x -axis is inverted for clarity.

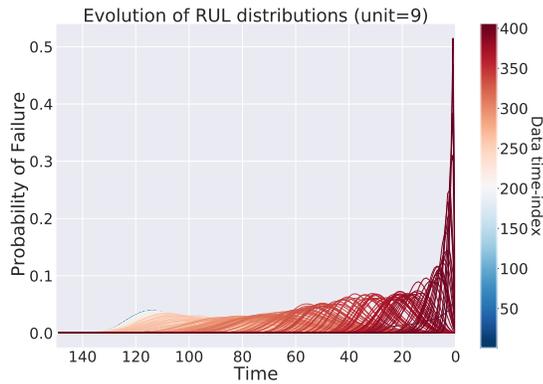


Figure 5. Evolution of Weibull distributions of unit 9 from FD003. Blue shades indicate the start of the unit’s trajectory and red shades the end. Note that the x -axis is inverted for clarity.

to see that the bi-objective HPO returned better scores for the RMSE and more “concentrated scores” for the uncertainty compared to the single-objective approach.

Regarding FD003 when examining Figures 3 and 9 we can see that the bi-objective approach returned, again, a larger number of hyperparameter configurations lying on the Pareto front (9 blue points on Figure 3) compared to the single-objective approach (7 blue points on Figure 9). Even though the number is marginally larger, this suggests, like previously, that the bi-objective approach might be more suitable for identifying a more diverse set of hyperparameters. In the bi-objective method, around 44% of the returned configurations result in RMSE lower than 20, with this number being around 57% in the single-objective approach. Nevertheless, we observe that the hyperparameter configurations from the bi-objective approach returned overall configurations with lower levels of uncertainty compared to the single-objective method. Specifi-

cally, more than 66% of the configurations on the bi-objective HPO result in uncertainty that is less than 2, with this number being around 43% in the single-objective HPO. Regarding the resulting hyperparameters’ performance (blue points) on the dedicated test set (red triangles), there are no apparent differences between the two methods’ topologies. Lastly, in the bi-objective method, the RMSE and uncertainty values of the dedicated test set lie in the range of [28.05, 68.01] and [0, 10.96], respectively. In the single-objective method, these ranges are [23.82, 50.76] and [0.14, 18.53], respectively, for the RMSE and uncertainty. This shows that for this dataset, the bi-objective method returned lower uncertainty values, but the single-objective approach returned RMSE values that lie in a more favorable range, thus indicating no clear winner.

From the previous results, we conclude that the usage of bi-objective HPO can reveal interesting trade-offs between the RMSE and uncertainty. Additionally, the results show that even though the bi-objective approach can return more configurations on the Pareto front, the single-objective HPO is also a good alternative for this task. The differences in the experimental findings between the two datasets might be justified by the fact that FD003 has 2 simulated fault conditions compared to FD001. In addition, we cannot rule out that the maximum allowable number of function evaluations or training epochs might have affected the findings, as more epochs might allow the network to learn more. More function evaluations of the HPO, on the other hand, will explore a larger part of the hyperparameter configuration space which might uncover better configurations.

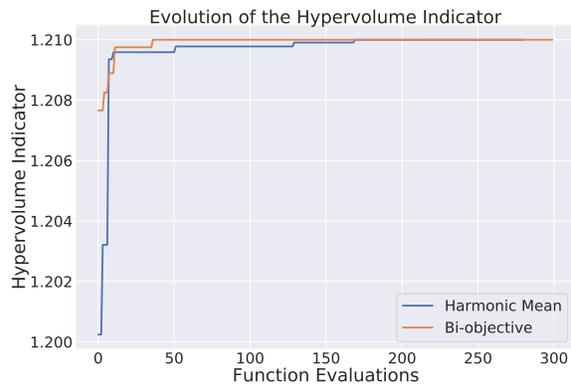


Figure 6. Evolution of the HVI of the bi-objective HPO and the single-objective HPO on FD001.

5.5. Application

Next, we will demonstrate how the proposed method can allow a more user-centric and interpretable approach to end-users (3rd contribution). For this application, we used the models which returned the lowest RMSE on the *dedicated test sets* of FD001 and FD003. These points are indicated with a green

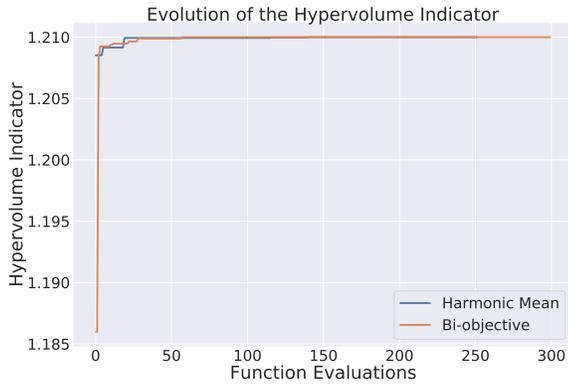


Figure 7. Evolution of the HVI of the bi-objective HPO and the single-objective HPO on FD003.

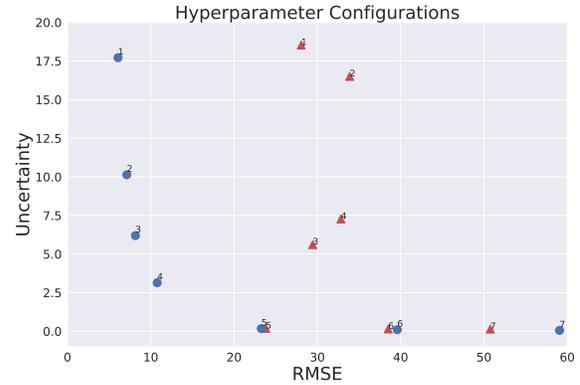


Figure 9. RMSE-UQ points corresponding to the hyperparameter configurations on FD003 using the harmonic mean approach. Blue circles are the Pareto front as calculated on the validation set. The red triangles are the points calculated on the dedicated test set.

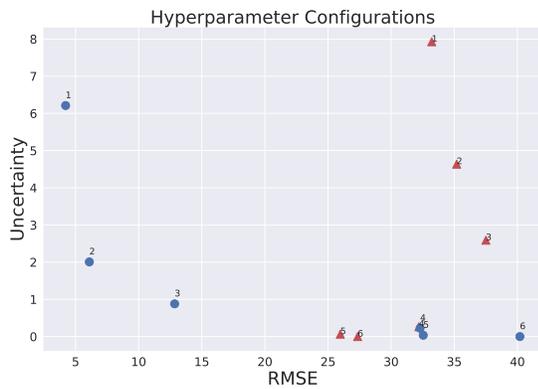


Figure 8. RMSE-UQ points corresponding to the hyperparameter configurations on FD001 using the harmonic mean approach. Blue circles are the Pareto front as calculated on the validation set. The red triangles are the points calculated on the dedicated test set.

marker on Figures 2 and 3. Specifically, since the trained network outputs the α and β parameters per input sample, the end-user can utilize this information to visualize, for example, the survival curves corresponding to each input sample, as well as other important information.

Survival curves are visualization methods from survival analysis that show the probability of an event *not* happening up to a point in time. In our case, this means that a failure has *not* occurred up to a point in time t (hence the asset will *survive* longer than t). A survival curve is defined as $1 - \text{CDF}$, where CDF stands for the cumulative distribution function (in this case, the Weibull’s CDF). For example in Figures 10 and 11 we plot the survival curves of test units 81, 4 from the FD001 dataset and test units 28, 3 from the FD003 dataset. For each test unit, we plot *all* the survival curves (shown within shaded areas for clarity) resulting from the multiple values of α and β that the network outputs through the MC Dropout, as well as the “median” curves that have as parameters the median val-

ues of the α s and β s, for a reference. This allows two things: the end-user can visually inspect the survival curves and, for instance, select a probability-of-survival threshold, based on one of them (e.g., the “median” curve), after which a unit should be maintained. Additionally, based on how wide the shaded areas are, the user can decide whether to employ the recommendation or proceed to further actions, such as further inspection by a field expert. For example, in Figure 11 the “median” survival curve of test unit 28 tells us that the probability of not having a failure up to time 100 from the current point in time (time 0) is about 80% and that this estimation is “more confident” compared to that of test unit 3, as the shaded area is less wide than the shaded area of test unit 3. Similarly, in Figure 10 the estimation of the survival curves of test unit 81 is “more confident” compared to that of test unit 4.

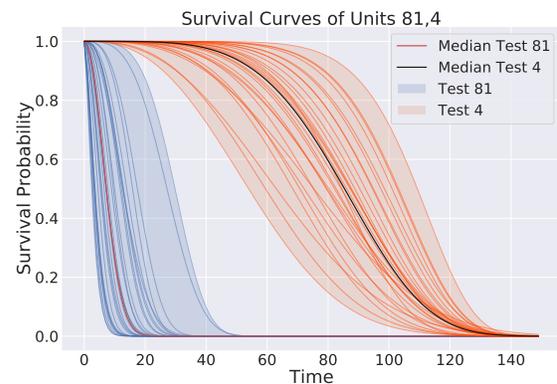


Figure 10. Survival curves of three units 81, 4 from FD001. The shaded areas include *all* the survival curves from the multiple passes through MC Dropout.

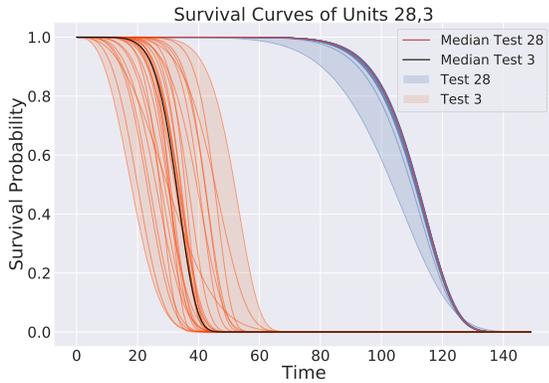


Figure 11. Survival curves of three units 28, 3 from FD003. The shaded areas include *all* the survival curves from the multiple passes through MC Dropout.

6. CONCLUSIONS AND OUTLOOK

In this work, we dealt with the remaining useful life (RUL) estimation using Bayesian deep learning (BDL) by taking into consideration the uncertainty of the estimate together with the predicted point estimate. We investigated the first, to our knowledge, usage of *bi-objective* hyperparameter optimization (HPO) that minimizes *simultaneously* the pointwise RMSE and the uncertainty. In this direction, we optimized together with the hyperparameters of the neural network (NN) the hyperparameters that govern the pre-processing steps, delivering thus, an *end-to-end*, data-driven, pipeline for the (offline) RUL estimation. We validated our approach on two subsets of the famous C-MAPSS dataset (A. Saxena & K. Goebel, 2008). We, further, demonstrated how survival curves can provide the end-user with information regarding the RUL and its confidence.

The experimental results indicate that, the bi-objective HPO might be more suitable for identifying a more diverse set of hyperparameter configurations compared to the single-objective HPO that aggregates the two objectives through the harmonic mean (HM). However, both methods reach the same hyper-volume indicator value of the Pareto front in, more or less, the same number of function evaluations and the findings did not indicate whether a method is more suitable for lower uncertainty or lower RMSE scores. Regarding the performance of the Pareto front configurations, when validated on the dedicated test sets, there was no clear winner between the two methods, although in the first examined case the RMSE values are better and the overall performance scores are clustered together. Overall, the results show that, for the examined cases, the bi-objective method is able to suggest more hyperparameter configurations and that the single-objective alternative is able to compete in terms of scores. This suggests that for a certain class of problems single-objective HPO methods are sufficient, allowing practitioners an ample selection of efficient

single-objective HPO methods.

Concerning the limitations of our work, due to the high computational costs of running the experiments multiple times no statistical significance tests are performed. Despite that fact, our methodology is experimentally sound and suggests an alternative approach for HPO in PHM. Furthermore, as indicated, we are aware that there is a current debate as to the validity of Monte Carlo Dropout being Bayesian (Osband et al., 2018). This could, in turn, make the corresponding predictive models problematic in support of reliable uncertainty quantification. As this work was mainly devoted to the usage of bi-objective hyperparameter optimization and user-centric approach, we have decided to address this *highly relevant but challenging issue* in future work. Future work should, in general, emphasize research on computationally efficient and accurate uncertainty quantification of DL models, as this will further open the road of AI applied in real-world applications.

Finally, we would be very interested in extending the bi-objective HPO to a many-objective context to add more objectives, such as run-time, to find a compromise between accuracy, uncertainty, and training time. The authors hope that multi-objective hyperparameter optimization methods become a new alternative, as it is not the case that a single objective method can always capture the conflicting interests that exist in real-world problems.

ACKNOWLEDGMENT

This work is part of the research programme Smart Industry SI2016 with project name CIMPLO and project number 15465, which is partly financed by the Netherlands Organisation for Scientific Research (NWO).

REFERENCES

- A. Saxena, & K. Goebel. (2008). *Turbofan engine degradation simulation data set*. NASA Ames Research Center, Moffett Field.
- Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., ... Nahavandi, S. (2021, December). A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges. *Information Fusion*, 76, 243–297. doi: 10.1016/j.inffus.2021.05.008
- Benker, M., Furtner, L., Semm, T., & Zaeh, M. F. (2021, October). Utilizing uncertainty information in remaining useful life estimation via Bayesian neural networks and Hamiltonian Monte Carlo. *Journal of Manufacturing Systems*, 61, 799–807. doi: 10.1016/j.jmsy.2020.11.005
- Biggio, L., Wieland, A., Chao, M. A., Kastanis, I., & Fink, O. (2021, April). Uncertainty-aware Remaining Useful Life predictor. *arXiv:2104.03613 [cs, stat]*.

- Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017, April). Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518), 859–877. doi: 10.1080/01621459.2017.1285773
- Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015, May). Weight Uncertainty in Neural Networks. *arXiv:1505.05424 [cs, stat]*.
- Caceres, J., Gonzalez, D., Zhou, T., & Drogue, E. L. (2021, October). A probabilistic Bayesian recurrent neural network for remaining useful life prognostics considering epistemic and aleatory uncertainties. *Structural Control and Health Monitoring*, 28(10). doi: 10.1002/stc.2811
- den Hertog, D., Kleijnen, J. P. C., & Siem, A. Y. D. (2006, April). The correct Kriging variance estimated by bootstrapping. *Journal of the Operational Research Society*, 57(4), 400–409. (Publisher: Taylor & Francis eprint: <https://doi.org/10.1057/palgrave.jors.2601997>) doi: 10.1057/palgrave.jors.2601997
- Emmerich, M. T. M., & Deutz, A. H. (2018, September). A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural Computing*, 17(3), 585–609. doi: 10.1007/s11047-018-9685-y
- Feurer, M., & Hutter, F. (2019). Hyperparameter Optimization. In F. Hutter, L. Kotthoff, & J. Vanschoren (Eds.), *Automated Machine Learning: Methods, Systems, Challenges* (pp. 3–33). Cham: Springer International Publishing. doi: 10.1007/978-3-030-05318-5_1
- Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd international conference on international conference on machine learning - volume 48* (p. 1050–1059). JMLR.org.
- Goodfellow, I., Yoshua Bengio, & Aaron Courville. (2016). *Deep Learning*. MIT Press.
- Govaers, F. (2019). *Introduction and Implementations of the Kalman Filter*. BoD – Books on Demand.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015, December). Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*.
- Hsu, C.-S., & Jiang, J.-R. (2018, April). Remaining useful life estimation using long short-term memory deep learning. In *2018 IEEE International Conference on Applied System Invention (ICASI)* (pp. 58–61). Chiba: IEEE. doi: 10.1109/ICASI.2018.8394326
- Hüllermeier, E., & Waegeman, W. (2021, March). Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods. *Machine Learning*, 110(3), 457–506. doi: 10.1007/s10994-021-05946-3
- Kalman, R. E. (1960, March). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1), 35–45. doi: 10.1115/1.3662552
- Kefalas, M., Baratchi, M., Apostolidis, A., van den Herik, D., & Bäck, T. (2021, June). Automated Machine Learning for Remaining Useful Life Estimation of Aircraft Engines. In *2021 IEEE International Conference on Prognostics and Health Management (ICPHM)* (pp. 1–9). Detroit (Romulus), MI, USA: IEEE. doi: 10.1109/ICPHM51084.2021.9486549
- Kim, M., & Liu, K. (2021, March). A Bayesian deep learning framework for interval estimation of remaining useful life in complex systems by incorporating general degradation characteristics. *IIEE Transactions*, 53(3), 326–340. doi: 10.1080/24725854.2020.1766729
- Kingma, D. P., & Ba, J. (2017, January). Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*.
- Kiureghian, A. D., & Ditlevsen, O. (2009, March). Aleatory or epistemic? Does it matter? *Structural Safety*, 31(2), 105–112. doi: 10.1016/j.strusafe.2008.06.020
- Kraus, M., & Feuerriegel, S. (2019, October). Forecasting remaining useful life: Interpretable deep learning approach via variational Bayesian inferences. *Decision Support Systems*, 125, 113100. doi: 10.1016/j.dss.2019.113100
- Krishna, M., & Baghaei, K. T. (2019). Recent Approaches in Prognostics: State of the Art. In *2019 International Conference on Artificial Intelligence (ICAI)* (pp. 358–365).
- Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018, may). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, 104, 799–834. doi: 10.1016/j.ymssp.2017.11.016
- Li, G., Yang, L., Lee, C.-G., Wang, X., & Rong, M. (2021, September). A Bayesian Deep Learning RUL Framework Integrating Epistemic and Aleatoric Uncertainties. *IEEE Transactions on Industrial Electronics*, 68(9), 8829–8841. doi: 10.1109/TIE.2020.3009593
- Li, X., Ding, Q., & Sun, J.-Q. (2018, April). Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172, 1–11. doi: 10.1016/j.res.2017.11.021
- Listou Ellefsen, A., Bjørlykhaug, E., Æsøy, V., Ushakov, S., & Zhang, H. (2019, March). Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture. *Reliability Engineering & System Safety*, 183, 240–251. doi: 10.1016/j.res.2018.11.027
- Malhotra, P., TV, V., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., & Shroff, G. (2016, August). Multi-Sensor Prognostics using an Unsupervised Health Index based on LSTM Encoder-Decoder. *arXiv:1608.06154 [cs]*.
- Martinsson, E. (2016). *Wtte-rnn: Weibull time to event recurrent neural network* (Unpublished master’s thesis). University of Gothenburg, Sweden.
- Nguyen, V. D., Kefalas, M., Yang, K., Apostolidis, A., Olhofer, M., Limmer, S., & Bäck, T. (2019). A Review: Prognostics and Health Management in Automotive and Aerospace. *International Journal of Prognostics*

- and Health Management*, 10(2), 35. doi: 10.36001/ijphm.2019.v10i2.2730
- Ordóñez, C., Sánchez Lasheras, F., Roca-Pardiñas, J., & Juez, F. J. d. C. (2019, January). A hybrid ARIMA–SVM model for the study of the remaining useful life of aircraft engines. *Journal of Computational and Applied Mathematics*, 346, 184–191. doi: 10.1016/j.cam.2018.07.008
- Osband, I., Aslanides, J., & Cassirer, A. (2018, November). Randomized Prior Functions for Deep Reinforcement Learning. *arXiv:1806.03335 [cs, stat]*.
- Peng, W., Ye, Z.-S., & Chen, N. (2020, March). Bayesian Deep-Learning-Based Health Prognostics Toward Prognostics Uncertainty. *IEEE Transactions on Industrial Electronics*, 67(3), 2283–2293. doi: 10.1109/TIE.2019.2907440
- Ramasso, E., & Saxena, A. (2014). Performance Benchmarking and Analysis of Prognostic Methods for CMAPSS Datasets. *International Journal of Prognostics and Health Management*, 5(2), 15. doi: <https://doi.org/10.36001/ijphm.2014.v5i2.2236>
- Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Cambridge, Mass: MIT Press.
- Sateesh Babu, G., Zhao, P., & Li, X.-L. (2016a). Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life. In S. B. Navathe, W. Wu, S. Shekhar, X. Du, X. S. Wang, & H. Xiong (Eds.), *Database Systems for Advanced Applications* (Vol. 9642, pp. 214–228). Cham: Springer International Publishing. (Series Title: Lecture Notes in Computer Science) doi: 10.1007/978-3-319-32025-0_14
- Sateesh Babu, G., Zhao, P., & Li, X.-L. (2016b). Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life. In S. B. Navathe, W. Wu, S. Shekhar, X. Du, X. S. Wang, & H. Xiong (Eds.), *Database Systems for Advanced Applications* (Vol. 9642, pp. 214–228). Cham: Springer International Publishing. (Series Title: Lecture Notes in Computer Science) doi: 10.1007/978-3-319-32025-0_14
- Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008, October). Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 International Conference on Prognostics and Health Management* (pp. 1–9). Denver, CO, USA: IEEE. doi: 10.1109/PHM.2008.4711414
- Si, X.-S., Wang, W., Hu, C.-H., & Zhou, D.-H. (2011). Remaining useful life estimation – A review on the statistical data driven approaches. *European Journal of Operational Research*, 213(1), 1 – 14. doi: <https://doi.org/10.1016/j.ejor.2010.11.018>
- Stein, B. v., Wang, H., & Back, T. (2019, July). Automatic Configuration of Deep Neural Networks with Parallel Efficient Global Optimization. In *2019 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–7). Budapest, Hungary: IEEE. doi: 10.1109/IJCNN.2019.8851720
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2014, February). Intriguing properties of neural networks. *arXiv:1312.6199 [cs]*.
- Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A., & Wu, B. (2006). Intelligent Fault Diagnosis and Prognosis for Engineering Systems.. doi: 10.1002/9780470117842
- Wang, B., Lei, Y., Yan, T., Li, N., & Guo, L. (2020, February). Recurrent convolutional neural network: A new framework for remaining useful life prediction of machinery. *Neurocomputing*, 379, 117-129. doi: <https://doi.org/10.1016/j.neucom.2019.10.064>
- Wang, H., van Stein, B., Emmerich, M., & Back, T. (2017, October). A new acquisition function for Bayesian optimization based on the moment-generating function. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 507–512). Banff, AB: IEEE. doi: 10.1109/SMC.2017.8122656
- Yang, F., Ren, H., & Hu, Z. (2019, May). Maximum Likelihood Estimation for Three-Parameter Weibull Distribution Using Evolutionary Strategy. *Mathematical Problems in Engineering*, 2019, 1–8. doi: 10.1155/2019/6281781
- Zhang, C., Lim, P., Qin, A. K., & Tan, K. C. (2017, October). Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2306–2318. doi: 10.1109/TNNLS.2016.2582798
- Zhao, Z., Wu, J., Wong, D., Sun, C., & Yan, R. (2020). Probabilistic Remaining Useful Life Prediction Based on Deep Convolutional Neural Network. *SSRN Electronic Journal*. doi: 10.2139/ssrn.3717738
- Zheng, S., Ristovski, K., Farahat, A., & Gupta, C. (2017, June). Long Short-Term Memory Network for Remaining Useful Life estimation. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)* (pp. 88–95). Dallas, TX, USA: IEEE. doi: 10.1109/ICPHM.2017.7998311
- Zhou, T., Droguett, E. L., Mosleh, A., & Chan, F. T. S. (2021, October). An uncertainty-informed framework for trustworthy fault diagnosis in safety-critical applications. *arXiv:2111.00874 [cs]*.
- Zitzler, E., Deb, K., & Thiele, L. (2000, June). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2), 173–195. doi: 10.1162/106365600568202

BIOGRAPHIES

Marios Kefalas currently pursues his Ph.D. in Predictive Maintenance and Optimization at the Leiden Institute of Advanced Computer Science (LIACS), Leiden University, The Netherlands. He received his BSc degree in Pure and Applied Mathematics at the Department of Mathematics, University of Athens, Greece, in 2015 and his MSc degree in Bioinformatics at LIACS, Leiden University, The Netherlands, in 2017. His research interests lie in Prognostics and Health Management, time-series application in industry and health, and Data Science.

Bas van Stein received his Ph.D. degree in Computer Science in 2018, from the Leiden Institute of Advanced Computer Science (LIACS), Leiden University, The Netherlands. From 2018 until 2021 he was a Postdoctoral Researcher at LIACS, Leiden University and he is currently an Assistant Professor at LIACS. His research interests lie in surrogate assisted optimisation, surrogate assisted neural architecture search and explainable AI techniques for industrial applications.

Mitra Baratchi received her Ph.D. degree in Computer Science in 2015, from the University of Twente, The Netherlands. In 2017, she joined the Leiden Institute of Advanced Com-

puter Science (LIACS), Leiden University, The Netherlands, as an Assistant Professor. Her research interests lie in machine learning for spatio-temporal and time-series data targeting various environmental and industrial applications.

Asteris Apostolidis received his Ph.D. degree in Computational Aerothermodynamics in 2015 from Cranfield University, UK. He worked for aircraft manufacturers, airlines and academic institutes and he is currently appointed as an Associate Professor at Amsterdam University of Applied Sciences. His interests include physics-based and data-driven methods for aircraft systems simulation, aircraft Maintenance Repair and Overhaul (MRO) and novel propulsion architectures.

Thomas Bäck (Fellow, IEEE) received the Diploma degree in Computer Science in 1990 and the Ph.D. degree in Computer Science in 1994, both from the University of Dortmund, Germany. Since 2002, he is Full Professor of Computer Science with the Leiden Institute of Advanced Computer Science (LIACS), Leiden University, The Netherlands. His research interests include evolutionary computation, machine learning, and their real-world applications, especially in sustainable smart industry and health.

Fault Detection in a Wind Turbine Hydraulic Pitch System Using Deep Autoencoder Extracted Features

Panagiotis Korkos¹, Jaakko Kleemola², Matti Linjama³, and Arto Lehtovaara⁴

^{1,3,4}*Tampere University, Tampere, 33014, Finland*
{firstname.lastname}@tuni.fi

²*Suomen Hyötytuuli Oy, Pori, 28601, Finland*
Jaakko.Kleemola@hyotytuuli.fi

ABSTRACT

A wind turbine is equipped with lots of sensors whose measurements are recorded by the supervisory control and data acquisition (SCADA) system and stored every 10 minutes. The pitch subsystem of a wind turbine is of critical importance as it presents the highest failure rate. Thus, selecting the most essential features from the SCADA system is performed in order to detect faults efficiently. In this study, a feature space of 49 features is available, referring to the condition of a hydraulic pitch system. The dimensionality of this feature space (original input space) is reduced using a Deep Autoencoder in order to extract latent information. The architecture of the Autoencoder is investigated regarding its efficiency on fault detection task. This way, effect of new extracted features on the performance of the classifier is presented. A Support Vector Machine (SVM) classifier is trained using a set of healthy (fault free) and faulty data, representing different kind of pitch system failures. The data are acquired from a wind farm of five 2.3MW fixed-speed wind turbines. The performance metric used to evaluate their effect on data is F1-score. Results show that SVM using new extracted feature by Autoencoder outperforms SVM classifier using the original feature set, underlining the power of Autoencoders to unveil latent information.

1. INTRODUCTION

Wind energy is the fastest developing renewable energy in the world, and especially in Europe. Based on the annual report of WindEurope, in 2021 the total installed wind power capacity was 236 GW (WindEurope, 2022). Wind turbine costs are strongly associated with the profitability and wind energy share in the energy production in daily basis. In particular, the total generation cost of wind energy is between 4.5 and 8.7 €cent/kWh in case of onshore wind turbine, but

the costs generated by Operation and Maintenance (O&M) is estimated to be 1-1.5 €cent/kWh (Blanco, 2009). Thus, O&M associated costs are very important and the only solution for consistent monitoring and maintenance is to accurately interpret the measurements. This interpretation is allowed through advanced data analysis techniques on the measurements of each wind turbine.

For that reason, each wind turbine is equipped with Supervisory Control and Data Acquisition (SCADA) system. SCADA system stores a plethora of measurements in a wind turbine ranging from environmental measurements to pressures and temperatures. Typically, measurements are stored in 10-min intervals even though they are sampled in higher frequency, e.g., 1 sec. Processing of SCADA signals has been a common strategy for a lot of windfarm operators, since it provides a cheap solution for wind turbine monitoring, avoiding the installation of more sensors.

A notable number of researchers have developed methodologies to process those SCADA signals for condition monitoring in wind turbines (Zaher, McArthur, Infield & Y. Patel, 2009; Chen, Zappala, Crabtree & Tavner, 2014; Tautz-Weinert & Watson, 2017; Yang, Court & Jiang, 2013). In addition, Stetco, Dinmohammadi, Zhao, Robu, Flynn, Barnes, Keane and Nenadic (2019) have summarized Machine Learning techniques that have been used in literature for wind turbine condition monitoring. Furthermore, more advanced techniques from the Deep Learning area have been the subject of the review in the study of Helbing and Ritter (2018), indicating the rise of Deep Learning for performing fault detection in wind turbines.

Regarding recent advancements in this application area, Convolutional Neural Network (CNN) have been widely used by researchers (Ulmer, Jarlskog, Pizza, Manninen & Goren Huber, 2020), as well as its variants such as convolutional neural network (CNN) and bidirectional gated recurrent unit (BiGRU) with attention mechanism (CNN-BiGRU-AM) (Xiang, Yang, Hu, Su & Wang, 2022), CNN

Panagiotis Korkos et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

with Long Short-Term Memory (LSTM) and attention mechanism (CNN-LSTM-AM) (Xiang, Wang, Yang, Hu & Su, 2021), convolutional neural networks (CNN) and gated recurrent unit (GRU) (Kong, Tang, Deng, Liu & Hana, 2020) and generative adversarial network (GAN) coupled with a temporal CNN (TCNN) (Afrasiabi, Afrasiabi, Parang, Mohammadi, Arefi & Rastegar, 2019). Additionally, different versions of autoencoders have been used like deep joint variational autoencoder (JVAE) for gearbox monitoring (Yang & Zhang, 2021), moving window stacked multilevel denoising AE (MW-SMDAE) (Chen, Li, Chen, Wang & Jiang, 2020) and sparse dictionary learning based adversarial variational auto-encoders (AVAE_SDL) (Liu, Teng, Wu, Wu, Liu & Ma, 2021). Finally, LSTM for gearbox monitoring has been applied by Qian, Tian, Kanfoud, Lee and Gan (2019).

The literature presented so far was mainly focused on the monitoring of the wind turbine, as a whole. The pitch system in wind turbines is very crucial for their operation since they present the highest failure rate and among the highest downtime according to several surveys (Wilkinson, Hendriks, Spinato, Harman, Gomez, Bulacio, Roca, Tavner, Feng, & Long, 2010; Carroll, McDonald, & McMillan, 2016; Ribrant, & Bertling, 2007). Therefore, it is considered the most critical subsystem and it needs to be monitored as effectively as possible. Pitch system monitoring has gained attention by several researchers. Chen, Matthews, and Tavner (2013, 2015) has implemented an a-priori adaptive neuro fuzzy inference system (APK-ANFIS) to monitor pitch system using as features five basic features (i.e., power output, wind speed, blade angle, rotor speed and motor torque for only the case of electric pitch system). Their study focused only on their average values. ANFIS has been used as well by Korkos, Linjama, Kleemola, and Lehtovaara (2022), investigating the effect of average and standard deviation values of the features mentioned in Chen et al. (2013, 2015). In addition, the novelty of their research was that their dataset contained a list of diverse pitch-system faults, referring to almost every kind of components. The same technique (ANFIS) was used in the studies of Schlechtingen, Santos, and Achiche (2013) and Schlechtingen and Santos (2014) in order to build normal behaviour models. Schlechtingen and Santos (2014) particularly used the model for hydraulic oil leakage, which is a common failure in the pitch system. Additionally, a pitch system fault, with no additional provided information, has been detected effectively using a multi-level-denoising autoencoder (MLD-AE) by Wu, Jiang, Wang, Xie, and Li (2019).

Apart from ANFIS, Support Vector Machines (SVM) have been used for fault detection. SVM classifiers have been developed by Leahy, Hu, Konstantakopoulos, Spanos, C.J., & Agogino (2016, 2018), whereas Hu, Leahy, Konstantakopoulos, Auslander, Spanos, and Agogino (2017) trained SVM classifiers in an enhanced feature set according

to domain knowledge. A variation of SVM, called asymmetric SVM, has been implemented by Wu, Su, Lu, and Rui (2015) to diagnose internal leakage of hydraulic cylinder. On the contrary, pitch-system fault detection has been dealt as a regression problem by Pandit and Infield (2019).

Finally, Gaussian Processes (GP) have been popular to some researchers dealing with pitch system faults. Pandit and Infield (2018) have trained their GP model using power curve, the rotor speed curve and the blade pitch angle curve as the feature set. Guo and Infield (2020) trained a multivariable power curve model with a modified Cholesky decomposition GP.

However, scientists have developed techniques to extract latent information from the SCADA signals in order to provide more enhanced information to wind turbine operators. These techniques belong to the broad area of the so-called dimensionality reduction techniques as well. In general, traditional Principal Component Analysis (PCA) (Jolliffe, 2002) has been applied in many fields, representing a linear transformation of input space. Additionally, nonlinear transformations have been applied to input space using the kernel trick in PCA, resulting in the kernel PCA (Smola, 1998). Nevertheless, the most advanced technique, arisen from the Deep Learning field, is Autoencoders (Goodfellow, Bengio, & Courville, 2016). Autoencoders are mainly a generalization of PCA, and they are based on neural network architectures. Denoising Autoencoders, which is a specific type of regularized Autoencoder has been used for dimensionality reduction techniques in wind turbines by Liu, Cheng, Kong, Wang, and Cui (2019) and Wu et al. (2019). But use of Autoencoders for dimensionality reduction has not been focused on pitch system monitoring. Thus, investigation of them is necessary and it has high potentials to provide more information about the condition of this subsystem to the operators. The extracted information will be also enhanced if the pitch faults, which are contained in the dataset, represent different kind of the most common faults. This is particularly interesting and adds up value in literature because studies in the past have failed to refer to specific types of faults that have been taken into account when setting up their dataset or have presented very limited information. Furthermore, the advantage of having more diverse faults is beneficial when performing identification of those types and that work will be realized in the future by the authors.

The objective of this study is to investigate the development of a Denoising Autoencoder (DAE), as a feature extraction technique, for fault detection of a wind turbine hydraulic pitch system. DAE makes use of nonlinear transformations of input space and its feature extraction potential is assessed through the performance of Support Vector Machine, which is used as classifier. This research has collected the most informative features for the hydraulic pitch system and the training dataset includes normal and faulty points derived from nine different faulty events. These faulty events include

diverse faults of every single component in the hydraulic pitch system, whose effect have not been investigated in earlier studies. The performance of the new latent dimensions on the classifier of SVM shows greater performance than using the original input space as input of SVM.

The paper is organized as follows. In Section 2, Deep Autoencoder for dimensionality reduction and feature extraction is described. In Section 3, the theory of SVM for classification problems is presented. Section 4 refers to the dataset of this research, which is referred to the hydraulic pitch system. Section 5 demonstrates the results, followed by the conclusions in the last section.

2. DEEP AUTOENCODER FOR DIMENSIONALITY REDUCTION

Autoencoders have been primarily used for dimensionality reduction tasks. Their clear advantage over other traditional dimensionality reduction techniques such as PCA is that they are based on nonlinear transformation of the input space. An autoencoder is composed of an encoder and a decoder. The encoder transforms the ambient space to a lower-dimensional space, in case of an undercomplete or to a higher dimensional if it is an overcomplete one. On the contrary, the decoder transforms the new feature space back to the original space.

Essentially, an autoencoder is a neural network which tries to learn the copying task of the input space. It requires only the input space and not the label, thus it belongs to unsupervised techniques. The encoder and decoder are typically nonlinear using several activation functions including sigmoid function, hyperbolic tangent function (tanh) or Rectified Linear Unit (ReLU).

More specifically, an encoder maps an input $x \in \mathbb{R}^m$ to a hidden representation h through the activation function f_s , shown in Eq.(1).

$$h = f_s(Wx + b) \quad (1)$$

where W is a $m \times m$ weight matrix and b is a bias vector. The decoder tries to reconstruct x from the latent representation, resulting in \hat{x} (Eq. (2)).

$$\hat{x} = f_s(W'h + b') \quad (2)$$

Where W' and b' are the parameters of the decoder in a similar way as in the encoder. An autoencoder is said to have tied weights if $W' = W^T$. The parameters of the autoencoder, represented shortly by $\theta = \{W, b\}, \theta' = \{W', b'\}$, are estimated after minimization of the average reconstruction error, demonstrated in Eq. (3).

$$\theta^*, \theta'^* = \underset{\theta, \theta'}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n L(x^{(i)}, \hat{x}^{(i)}) \quad (3)$$

The loss function L is the traditional mean squared error $L(x, \hat{x}) = \|x - \hat{x}\|^2$.

Even though an autoencoder deals with the copying task of its input to its output, exact reconstruction is useless and no new latent information is extracted. In addition, if both the encoder and decoder functions are given too much capacity, it fails to learn anything useful. That is the reason why researchers suggested regularized autoencoders, which additionally provide sparsity of the representation, smallness of the derivative of the representation and robustness to noise and missing inputs (Goodfellow et al, 2016). Such regularized autoencoders are sparse autoencoders and denoising autoencoders.

Denoising Autoencoders (DAE) are similar to the traditional autoencoders, but the input of them is a corrupted version of original input space. Furthermore, the end goal is to predict the original input and not the corrupted one. Consequently, before implementing autoencoder, the input is corrupted by either adding Gaussian noise or salt-and-pepper noise or masking noise (Vincent, Larochelle, Bengio, & Manzagol, 2008). In other words, the input of a DAE will be the corrupted \tilde{x} and not x , and the loss function is the L^2 norm between the reconstruction of corrupted datapoints and original datapoints.

3. SVM AS CLASSIFIER

Support Vector Machines (SVM) (Cortes & Vapnik, 1995) have gained a lot of attention from 2000 onwards due to its ability to provide better classification performance, compared to Artificial Neural Networks. However, it can be also used for regression problems. In particular, SVMs nonlinearly map the input space into a higher-dimensional space and then a linear decision boundary is set to separate the classes. Therefore, it may seem that a linear decision line has been constructed, but in reality, this line is nonlinear in the original space. Finally, the decision boundary is based on the support vectors, which are essentially a small amount of datapoints that allow to define the best separation boundary between two classes.

SVM is given in Eq. (4), which clearly shows its dependence on a nonlinear transformation φ .

$$f(x) = u^T \varphi(x) + d \quad (4)$$

where $\varphi(x)$ is the nonlinear transformation of the input space to the high-dimensional feature space. The output of SVM is not probabilities, but the class label. In other words, if f is positive, SVM predicts the positive class and if f is negative, it predicts the negative class.

The u and d parameters are determined by minimizing the regularized risk function. However, most of the times some of the datapoints are allowed to be misclassified, leading to soft margin SVM. Soft margin SVM is given by minimizing

the dual Lagrangian \tilde{L} function, shown in Eq. (5), where some of the datapoints are allowed to be misclassified.

$$\tilde{L}(a) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m y_n y_m K(x_n, x_m) \quad (5)$$

where $K(x_n, x_m) = \varphi(x_n)^T \varphi(x_m)$, that represents a kernel, thus leading to nonlinear SVM. a_n are non-negative Lagrangian multipliers, which define the final solution for u and d shown in Eq. (6) and Eq. (7) respectively.

$$u = \sum_{n=1}^N a_n y_n \phi(x_n) \quad (6)$$

$$d = \frac{1}{N_{\mathcal{M}}} \sum_{n \in \mathcal{M}} \left(y_n - \sum_{m \in \mathcal{S}} a_m y_m K(x_n, x_m) \right) \quad (7)$$

where \mathcal{M} represent the set of indices of data points where $0 < a_n < C$, C is the regularization constant and y_n are the target values of the input x_n .

Common kernels for kernel SVM are polynomial kernels, sigmoid kernel and Radial Basis Function (RBF) kernels (Eq. (8)), whose hyperparameter γ is half of the variance of the standard normal density.

$$K(x_n, x_m) = \exp(-\gamma \|x_n - x_m\|^2) \quad (8)$$

4. DATASET

This study makes use of 10-year long available data, derived from the SCADA system of a windfarm in western Finland. The studied windfarm includes five wind turbines of 2.3 MW, which are fixed-speed and have a hydraulic pitch system. These SCADA data include average, standard deviation, maximum and minimum of several measurements stored in 10-min intervals. Nevertheless, the objective of this research is fault detection in the hydraulic pitch system, thus the most effective parameters have been selected which have the biggest impact on its operation.

These features have been preprocessed and labelled according to Korkos et al (2022). Then, features have been normalized using Min-Max normalization (Eq. (9)). The normalized values would be in the range between 0 and 1.

$$x_{new}^i = \frac{x^i - x_{min}}{x_{max} - x_{min}} \quad (9)$$

where, x^i and x_{new}^i are the original and normalized feature respectively and x_{min} and x_{max} are the minimum and maximum values of each feature.

Table 1. SCADA features and their short names demonstrates the list of features that were used as input at the dimensionality reduction technique. Their names are mentioned using shortened form followed by {"_mean", "_stdev", "_max", "_min"}. However, only gust wind speed contains a single value, instead of the statistical quantities mentioned before. For example, if maximum value of power output is mentioned, the shortened name will be "PO_max". In total, the original feature space is 49-dimensional.

Table 1. SCADA features and their short names

Name	Description	Blade
RS	Rotor speed	-
BAA	Blade angle A	A
BAB	Blade angle B	B
BAC	Blade angle C	C
WS	Wind speed	-
PO	Power output	-
Gust_WS	Gust wind speed	-
HPrA	Hub Pressure A	A
HPrB	Hub Pressure B	B
HPrC	Hub Pressure C	C
HydP	Hydraulic Pressure	-
AmbT	Ambient Temp.	-
HubT	Hub Temp.	-

This study collected a dataset which contains normal and faulty operation datapoints. More specifically, faulty dataset contains data when different kind of events of faults were occurred. In particular, Table 2 shows the nine pitch events that have been taken into account for this study. For normal data points the label has been assigned to zero and for faulty data points the label is one. The data are owned by Suomen Hyötytuuli Oy and are not publicly available due to confidentiality reasons.

Table 2. Event list

No	Pitch event
1	Hydraulic hoses and oils replacement
2	Hub oil leakage + Hyd. Oil replacement + Bl. valve 6 replacement
3	Block replacement at blade B (No3)

4	Block leakage in blade B(No1)
5	Replacement of A- blade valve 102 (No3)
6	Replacement of A, B, C- blade valve 116 (No3)
7	Nitrogen accumulator (No 4) replacement of Blade A (No5)
8	Blade tracking error during stop/operation of Blade A (No1)
9	Replacement of hyd. cylinder (No2)

5. RESULTS AND DISCUSSION

New features have been extracted using a Denoising Autoencoder (DAE). Autoencoders have the advantage to use nonlinear transformation of input space. Thus, they belong to nonlinear dimensionality reduction techniques. This study investigated different architectures of DAEs. These architectures are presented on the Table 3, as well as their activation functions. If n is the dimension of original dataset, $n = 49$ for this study which is the dimension of both input and output layer.

Table 3. Different architectures of DAEs under investigation

No	Architecture	Activation Function
1	$[n,64,32,16,8,16,32,64,n]$	ReLU
2	$[n,32,32,16,8,16,32,32,n]$	ReLU
3	$[n,32,32,16,8,16,32,32,n]$	sigmoid
4	$[n,32,8,32,n]$	sigmoid
5	$[n,32,24,16,10,6,10,16,24,32,n]$	sigmoid

The best architecture was achieved by $[n,32,32,16,8,16,32,32,n]$ (Figure 1) using sigmoid function as activation function. In addition, Mean Squared Error (MSE) was chosen as loss function and the optimization algorithm was Adam algorithm. The corruption of input was selected to be a Gaussian noise. Gaussian noise was represented by the standard normal distribution $N(0,1)$ multiplied by 0.02. This multiplier has been chosen after appropriate tuning.

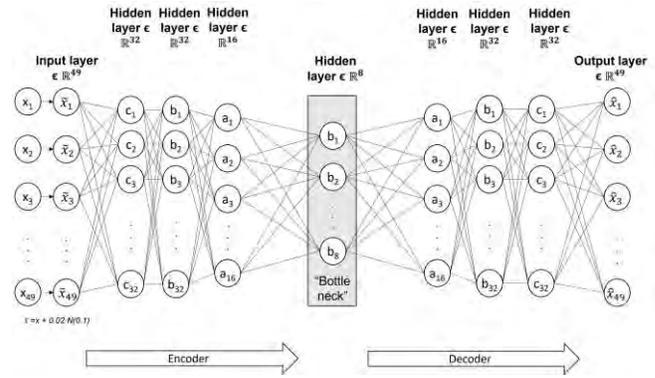


Figure 1. Denoising Autoencoder (DAE) $[n,32,32,16,8,16,32,32,n]$ architecture

Figure 2 demonstrates a two-dimensional representation of 8D latent space. T-distributed Stochastic Neighbor Embedding has been applied to the new extracted features (8D) in order to provide a visualization of them. Figure 2 shows that the two classes can be clearly separated. Thus, features extracted by the developed DAE, shown in Figure 1, really extracts hidden information and helps to separate the two classes more clearly.

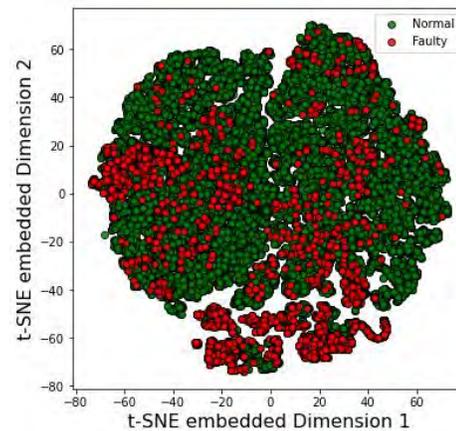


Figure 2. Two of the new extracted features using Autoencoder $[n,32,32,16,8,16,32,32,n]$

After the reduction of the dimensions and the extraction of the new features, Support Vector Machine classifier was trained in order to perform the fault detection task. Hyperparameter tuning of SVM has been performed through cross validation between the regularization constant C , type of kernel and hyperparameter γ , in case of Radial Basis Function (RBF) kernel. More specifically, this research investigated values of C in the list $\{0.01, 0.1, 1, 10, 100, 1000\}$ (being either linear or RBF kernel) as well as γ values in the list $\{0.1, 1, 10, 50, 100, 500\}$ should the kernel is RBF.

Dataset has been split in two parts, i.e., 80% for training and 20% for testing. Training dataset is separated in training dataset and validation set during cross-validation process in

order to determine the hyperparameters. The final training of the SVM classifier has been done in the whole training set. The performance of the classifier was assessed based on the F1-score, shown in Eq. (10). This performance metric was chosen instead of other such as accuracy because normal operation class represents the vast majority of the datapoints and a correct evaluation requires to take into account that missed faulty points will be shown at the performance metric.

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (10)$$

where TP represent True Positive, meaning that the faulty points (label “1”) were truly detected. The same notion is followed for FP (False Positive) and FN (False Negative), whose actual label was “0” and “1” respectively, but the opposite class was predicted.

Table 4 summarizes the results of F1-score when performing 3-fold cross validation for every combination of kernel, C and γ values. F1-scores are given as an average value during calculation of it in the 3-fold datasets and standard deviation is presented within parentheses. Best linear SVM model is acquired for C value of 1000. In contrast, the highest F1-scores using RBF SVM is received when using C = 1000 and $\gamma = 10$, which has the best performance in the validation set among all investigated classifiers.

Table 4. Best F1-scores of SVM for different pairs of C, γ and kernel

F1-score	C	γ	kernel
0.731 (+/-0.013)	1000	-	linear
0.582 (+/-0.018)	0.01	10	RBF
0.816 (+/-0.021)	0.1	50	RBF
0.917 (+/-0.006)	1	100	RBF
0.936 (+/-0.007)	10	50	RBF
0.937 (+/-0.004)	100	10	RBF
0.938 (+/-0.005)	1000	10	RBF

Therefore, when using the developed Denoising Autoencoder, as feature extractor shown in Figure 1, the SVM performance for C = 1000 and $\gamma = 10$ is 0.9457%, according to F1-score. This study uses as benchmark the SVM performance when using only the original features. Benchmark’s performance is 0.8538%, thus the developed DAE provides increase of 10.8%. This result outperforms the performance of Adaptive Neuro Fuzzy Inference system (ANFIS) presented in Korkos et al. (2022). Results from other similar studies could not be directly compared to the present one. The reason is the dataset variability since each researcher uses a different dataset. However, Leahy et al. (2016) attained 65% F1-score for fault detection task without

mentioning the details of the faults. Moreover, Hu et al. (2017) achieved 90% F1-score by increasing their feature set, which contained only the original SCADA features. Finally, APK-ANFIS model, developed by Chen et al. (2015), achieved 50% of F1-score for fixed-speed wind turbines using some pitch faults, providing no information about them. Consequently, the attained F1-score of the present study leads to the conclusion that Denoising Autoencoders are very powerful at extracting useful information out of the dataset.

6. CONCLUSION

In this paper, a Denoising Autoencoder (DAE) has been developed to extract hidden information that will contribute to more efficient monitoring of wind turbine hydraulic pitch system. The efficiency of DAE has been evaluated based on the performance of a Support Vector Machines (SVM) classifier, which uses the new extracted features as input. More specifically, the original feature set had been 49-dimensional, including from environmental parameters to several pressures in the pitch system. Hence, the nonlinear transformations, employed by the developed DAE, attained 0.9457%, which was 10.8% better than the case of SVM using directly the original feature set. As a result, pitch system, which is crucial for a wind turbine, can be monitored more effectively and accurately. Additionally, those extracted features may be used in future studies for diagnosing each fault separately. That information would provide great assistance to wind turbine operators and will lower maintenance costs. Possible other classifiers, from the Deep Learning field, may be investigated in the future such as 1D Convolutional Neural Network or Long Short-Term Memory network (LSTM).

ACKNOWLEDGEMENT

This research was funded by the Doctoral School of Industry Innovations (DSII) of Tampere University and Suomen Hyötytuuli Oy.

REFERENCES

Afrasiabi, S., Afrasiabi, M., Parang, B., Mohammadi, M., Arefi, M. M., & Rastegar, M. (2019). Wind turbine fault diagnosis with Generative-Temporal Convolutional Neural Network, *2019 IEEE International Conference on Environment and Electrical Engineering and 2019 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, pp. 1-5. doi: 10.1109/EEEIC.2019.8783233

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*, New York: Springer Science+Business Media, LLC.

Blanco, I. (2009). The economics of wind energy, *Renewable and Sustainable Energy Reviews*, vol. 13, pp. 1372-1382. doi: 10.1016/j.rser.2008.09.004

- Carroll, J., McDonald, A., & McMillan, D. (2016). Failure rate, repair time and unscheduled O & M cost analysis of offshore wind turbines, *Wind Energy*, vol. 19, pp. 1107-1119. doi: 10.1002/we.1887
- Cortes, C., Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, vol. 20, pp. 273-297. doi: 10.1007/BF00994018
- Chen, J., Li, J., Chen, W., Wang, Y., & Jiang, T. (2020) Anomaly detection for wind turbines based on the reconstruction of condition parameters using stacked denoising autoencoders, *Renewable Energy*, vol. 147, pp. 1469-1480. doi: 10.1016/j.renene.2019.09.041
- Chen, B., Matthews, P.C., & Tavner, P.J. (2013) Wind turbine pitch faults prognosis using a-priori knowledge-based ANFIS, *Expert Systems with Applications*, vol. 40, pp. 6863-6876. doi: 10.1016/j.eswa.2013.06.018
- Chen, B., Matthews, P.C., & Tavner, P.J. (2015) Automated on-line fault prognosis for wind turbine pitch systems using supervisory control and data acquisition, *IET Renewable Power Generation*, vol. 9, pp. 503-513. doi: 10.1049/iet-rpg.2014.0181
- Chen, B., Zappala, D., Crabtree, C.J., & Tavner, P.J. (2014) *Survey of commercially available SCADA data analysis tools for wind turbine health monitoring*. Technical Report. Durham University School of Engineering and Computing Sciences
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*, Cambridge, MA: MIT Press
- Guo, P., & Infield, D. (2020). Wind turbine power curve modeling and monitoring with Gaussian process and SPRT, *IEEE Trans. Sustain. Energy*, vol. 11, pp. 107-115. doi: 10.1109/TSSTE.2018.2884699
- Helbing, G., & Ritter, M. (2018). Deep Learning for fault detection in wind turbines, *Renewable and Sustainable Energy Reviews*, vol. 98, pp. 189-198. doi: 10.1016/j.rser.2018.09.012
- Hu, R.L., Leahy, K., Konstantakopoulos, I.C., Auslander, D.M., Spanos, C.J., & Agogino, A.M. (2017). Using domain knowledge features for wind turbine diagnostics, *Proceedings of 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. pp. 300-305. doi: 10.1109/ICMLA.2016.172
- Kong, Z., Tang, B., Deng, L., Liu W., & Hana, Y. (2020). Condition monitoring of wind turbines based on spatio-temporal fusion of SCADA data by convolutional neural networks and gated recurrent units, *Renewable Energy*, vol. 146, pp. 760-768. doi: 10.1016/j.renene.2019.07.033
- Korkos, P., Linjama, M., Kleemola, J., & Lehtovaara, A. (2022). Data annotation and feature extraction in fault detection in a wind turbine hydraulic pitch system. *Renewable Energy*, vol. 185, pp. 692-703. doi: 10.1016/j.renene.2021.12.047
- Leahy, K., Hu, R.L., Konstantakopoulos, I.C., Spanos, C.J., & Agogino, A.M. (2016). Diagnosing wind turbine faults using machine learning techniques applied to operational data, *2016 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pp. 1–8. doi: 10.1109/ICPHM.2016.7542860
- Leahy, K., Hu, R.L., Konstantakopoulos, I.C., Spanos, C.J., Agogino, A.M., & O’Sullivan, D.T.J. (2018). Diagnosing and predicting wind turbine faults from SCADA data using support vector machines, *International Journal of Prognostics and Health Management*, vol. 9 (1), pp. 1-11. doi: 10.36001/ijphm.2018.v9i1.2692
- Liu, Y., Cheng, H., Kong, X., Wang, Q., & Cui, H. (2019). Intelligent wind turbine blade icing detection using supervisory control and data acquisition data and ensemble deep learning, *Energy Science Engineering*, vol. 7, pp. 2633-2645. doi: 10.1002/ese3.449
- Liu, X., Teng, W., Wu, S., Wu, X., Liu, Y., & Ma, Z. (2021), Sparse dictionary learning based adversarial variational auto-encoders for fault identification of wind turbines, *Measurement*, vol. 183. doi: 10.1016/j.measurement.2021.109810
- Pandit, R.K., & Infield, D. (2018). Gaussian process operational curves for wind turbine condition monitoring, *Energies*, vol. 11 (7). doi: 10.3390/en11071631
- Pandit, R.K., & Infield, D. (2019). Comparative assessments of binned and support vector regression-based blade pitch curve of a wind turbine for the purpose of condition monitoring, *International Journal of Energy and Environmental Engineering*, vol. 10, pp. 181-188. doi: 10.1007/s40095-018-0287-3
- Qian, P., Tian, X., Kanfoud, J., Lee, J.L.Y., & Gan, T.H. (2019). A Novel Condition Monitoring Method of Wind Turbines Based on Long Short-Term Memory Neural Network, *Energies*, vol. 12 (18). doi: 10.3390/en12183411
- Ribrant, J., & Bertling, L.M. (2007) Survey of failures in wind power systems with focus on Swedish wind power plants during 1997-2005, *IEEE Trans. Energy Convers.*, vol. 22, pp. 167-173. doi: 10.1109/PES.2007.386112
- Schlechtingen, M., Santos, I.F., & Achiche, S. (2013) Wind turbine condition monitoring based on SCADA data using normal behavior models. Part 1: System description, *Applied Soft Computing*, vol. 13, pp. 259-270. doi: 10.1016/j.asoc.2012.08.033
- Schlechtingen, M., & Santos, I.F. (2014) Wind turbine condition monitoring based on SCADA data using normal behavior models. Part 2: Application examples, *Applied Soft Computing*, vol. 14, pp. 447-460. doi: 10.1016/j.asoc.2013.09.016
- Stetco, A., Dinmohammadi, F., Zhao, X., Robu, V., Flynn, D., Barnes, M., Keane, J., & Nenadic, G. (2019) Machine learning methods for wind turbine condition monitoring: A review, *Renewable Energy*. vol. 133, pp. 620-635. doi:10.1016/j.renene.2018.10.047
- Tautz-Weinert, J., & Watson, S.J., (2017). Using SCADA data for wind turbine condition monitoring - A review,

- IET Renewable Power Generation*, vol. 11, pp. 382-394. doi: 10.1049/iet-rpg.2016.0248
- Ulmer, M., Jarlskog, E., Pizza, G., Manninen, J., & Goren Huber, L. (2020). Early Fault Detection Based on Wind Turbine SCADA Data Using Convolutional Neural Networks. *PHM Society European Conference*, vol. 5(1), 9. doi: 10.36001/phme.2020.v5i1.1217
- Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.A. (2008). Extracting and composing robust features with denoising autoencoders, *Proceedings 25th International Conference on Machine Learning* (pp. 1096-1103), Helsinki, Finland. doi: 10.1145/1390156.1390294
- Wilkinson, M., Hendriks, B., Spinato, F., Harman, K., Gomez, E., Bulacio, H., Roca, J., Tavner, P., Feng, Y., & Long, H. (2010). Methodology and results of the Reliawind reliability field study, *European Wind Energy Conference Exhibition, EWEC 2010*. April 20-23, Warsaw, Poland.
- WindEurope. (2022). *Wind energy in Europe - 2021 Statistics and the outlook for 2022-2026*, Annual report, Brussels, Belgium
- Wu, X., Jiang, G., Wang, X., Xie, P., & Li, X. (2019). A Multi-Level-Denoising Autoencoder approach for wind turbine fault detection, *IEEE Access*, vol. 7, pp. 59376-59387. doi: 10.1109/ACCESS.2019.2914731
- Wu, X., Su, R., Lu, C., & Rui, X. (2015). Internal leakage detection for wind turbine hydraulic pitching system with computationally efficient adaptive asymmetric SVM, *Proceedings of 2015 34th Chinese Control Conf.*, pp. 6126-6130, July 28-30, Hangzhou, China. doi: 10.1109/ChiCC.2015.7260599
- Xiang, L., Wang, P., Yang, X., Hu, A., & Su, H. (2021). Fault detection of wind turbine based on SCADA data analysis using CNN and LSTM with attention mechanism, *Measurement*, vol. 175. doi: 10.1016/j.measurement.2021.109094
- Xiang, L., Yang, X., Hu, A., Su, H., & Wang, P. (2022). Condition monitoring and anomaly detection of wind turbine based on cascaded and bidirectional deep learning networks, *Applied Energy*, vol. 305, doi: 10.1016/j.apenergy.2021.117925
- Yang, W., Court, R., & Jiang, J., (2013). Wind turbine condition monitoring by the approach of SCADA data analysis, *Renewable Energy*, vol. 53, pp. 365-376. doi: 10.1016/j.renene.2012.11.030
- Yang, L., & Zhang, Z. (2021). Wind turbine gearbox failure detection based on SCADA data: A Deep Learning-based approach, *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-11, doi: 10.1109/TIM.2020.3045800
- Zaher, A., McArthur, S.D.J., Infield, D.G., & Patel, Y. (2009) Online wind turbine fault detection through automated SCADA data analysis. *Wind Energy*, vol. 12, pp. 574-593. doi: 10.1002/we.319

***iVRIDA*: intelligent Vehicle Running Instability Detection Algorithm for high-speed rail vehicles using Temporal Convolution Network – A pilot study**

Rohan R Kulkarni, Rocco Libero Giossi, Prapanpong Damsongsaeng, Alireza Qazizadeh and Mats Berg

Department of Engineering Mechanics, KTH Royal Institute of Technology, SE 100 44, Stockholm, Sweden

rohank@kth.se

ABSTRACT

Intelligent fault identification of rail vehicles from onboard measurements is of utmost importance to reduce the operating and maintenance cost of high-speed vehicles. Early identification of vehicle faults responsible for an unsafe situation, such as the instable running of high-speed vehicles, is very important to ensure the safety of operating rail vehicles. However, this task is challenging because of the nonlinear dynamics associated with multiple subsystems of the rail vehicle. The task becomes more challenging with only accelerations recorded in the carbody where, nevertheless, sensor maintenance is significantly lower compared to axlebox accelerometers. This paper proposes a Temporal Convolution Network (TCN)-based intelligent fault detection algorithm to detect rail vehicle faults. In this investigation, the classifiers are trained and tested with the results of numerical simulations of a high-speed vehicle (200 km/h). The TCN based fault classification algorithm identifies the rail vehicle faults with 98.7% accuracy. The proposed method contributes towards digitalization of rail vehicle maintenance through condition-based and predictive maintenance.

1. INTRODUCTION

Vehicle hunting motion (running instability) is an important phenomenon in vehicle-track dynamic interaction and typically appears at a fairly high vehicle speed and on a straight track or in large-radius curves. The running instability is an intrinsic behaviour of a vehicle system that is dependent on the health of the vehicle and track subsystems. The foremost reasons of running instability are poor vehicle yaw dampers, too soft primary suspension in the horizontal plane or poor wheel-rail interface geometry. Vehicle hunting is a safety concern and can also cause passenger discomfort. The European Standard *EN 14363:2016+AI* (2019) standard

specifies the methods to measure vehicle running instability in the vehicle certification phase. However, these methods are not suitable for continuous health monitoring of the vehicle and track subsystems which influences the running instability of the vehicle. Gasparetto et al., (2013) employ Random Decrement Technique to extract the vehicle's hunting frequency and residual damping from bogie frame accelerations. These signal-based features are fed into k Nearest Neighbor (kNN) and Artificial Neural Network (ANN) fault classifiers to diagnose the reason behind the observed vehicle running instability, mainly vehicle-based faults. Ning et al., (2018), propose data-driven fault classifiers combined with data fusion of multiple bogie frame accelerations for diagnostics of vehicle hunting. The authors employ Empirical Mode Decomposition (EMD) and Sample Entropy (SE) methods to extract features associated with small amplitude hunting and incorporate them into Support Vector Machine (SVM) classifier as fault identifiers. Zeng et al., (2020) use a phase-space reconstruction algorithm to extract signal-based features to estimate the state variables periodicity in the nonlinear dynamic system and detect hunting based on axlebox accelerations. Kulkarni et al., (2019), deployed two classifiers (i.e., linear SVM and Gaussian SVM) for the Fault Detection and Isolation (FDI) of yaw dampers of high-speed trains. The simulation results showed that both classifiers could identify the faulty yaw dampers well. Moreover, the Gaussian SVM classifier performed slightly better in the training and testing phases, while it had a higher risk of overfitting the current dataset. Overall, the results showed the ability of the data-driven approach to be used for the FDI of railway vehicle suspension faults. The articles above, mainly extract features from axlebox acceleration or bogie frame acceleration and mainly use traditional machine learning algorithms. Moreover, these studies do not focus on intelligent fault identification of vehicle running instability.

The main objective of the present study is to detect vehicle running instability and identify the root causes from carbody floor acceleration using two different methods. Namely,

Rohan Kulkarni et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Dynamic Mode Decomposition (DMD) (Brunton & Kutz, 2019) and Temporal Convolutional Neural Network (TCN) (Lea et al., 2016). The DMD method accurately estimates the eigenfrequencies and eigenmodes of the system. In recent times, a TCN is proposed which shows excellent abilities in solving sequential problems such as analysing time series data and outperforms Recurrent Neural Network (RNN) models. Thus, TCN is deployed to identify the root causes of observed vehicle running instability in this investigation. The iVRIDA algorithm is described in the next subsection which is followed by results and conclusions.

2. iVRIDA ALGORITHM

2.1. Algorithm Schematic

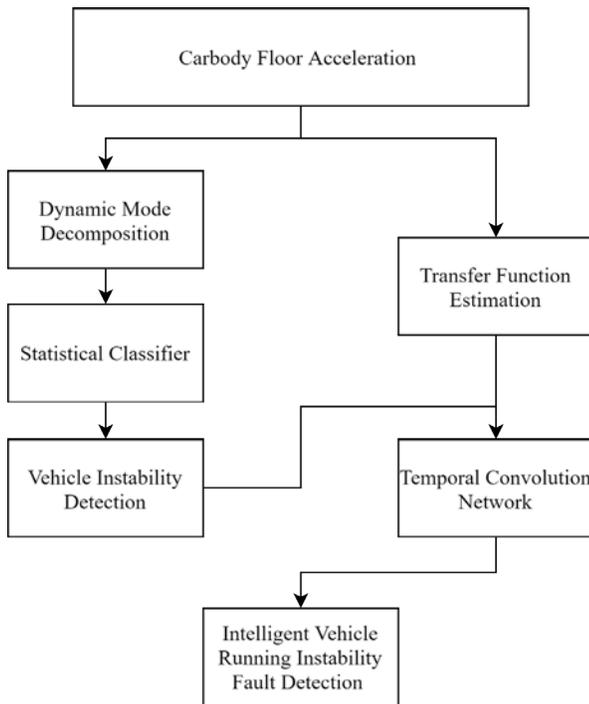


Figure 1 Schematic of iVRIDA algorithm

The proposed iVRIDA algorithm for vehicle running instability detection and root cause identification from carbody floor acceleration is illustrated in Figure 1. The algorithm utilizes two data-driven methods namely DMD and TCN aiming at detecting the vehicle instability and identifying root cause of the same. The hunting or vehicle running instability detection algorithm is implemented based on a binary classification method using outputs from DMD of carbody floor accelerations. Besides, the root causes identification is a classical multi-class classification problem and TCN is deployed on transfer functions between track and carbody floor.

2.2. Vehicle Running Instability Detection with DMD

The vehicle-track is a system where the nonlinearities mainly lie in the contact between the wheel and the rail. While hunting, a limit cycle is reached, and the system starts to oscillate at a certain frequency and with a precise mode shape. In this condition, an almost stable cycle can be detected. Thus, during hunting, the vehicle (bogies plus carbody) can be considered linear. This first assumption makes it possible to apply the DMD algorithm. Non-linearities are not expected in the time and spatial domains.

The DMD algorithm is chosen because it is a fast and accurate algorithm with which the eigenfrequencies and eigenmodes of the system can be detected. It is convenient in hunting detection due to the order in which the results are sorted, namely by energy content. In fact, in hunting motion, essentially only one mode will be excited. This mode will be the one with the highest energy content. It will be sufficient to consider the this mode.

The DMD algorithm assumes linear relation in time and space for the selected signals. The relation between the time t_m and the previous one t_{m-1} can thus be defined,

$$X(t_2, \dots, t_m) = AX(t_1, \dots, t_{m-1}) \rightarrow X_2 = AX_1, \quad (1)$$

where, X_1 is the $n \times (m-1)$ matrix representing the state of the system at instances from t_1 to $t_{(m-1)}$, X_2 is the $n \times (m-1)$ matrix representing the state of the system at instances from t_2 to t_m and A is the state matrix. Here, n is the number of sensors used and m is the number of time steps. Applying the reduced Singular Value Decomposition (SVD) of the matrix X_1 with reduced order r ,

$$X_1 = U_r \Sigma_r V_r^*, \quad (2)$$

it is possible to estimate the state matrix A to the reduced order r ,

$$\hat{A} = U_r^* X_2 V_r \Sigma_r^{-1}. \quad (3)$$

If the eigenvalue problem is solved for the matrix \hat{A} ,

$$\hat{A}W = W\Lambda, \quad (4)$$

It is now possible to determine the mode shapes Φ and the eigenfrequencies f of the system,

$$\Phi = X_2 V_r \Sigma_r^{-1} W, f = \log(\Lambda) f_s, \quad (5)$$

where, f_s is the sampling frequency.

During the analysis, the DMD algorithm with second-order reduction applied to carbody acceleration was able to identify correctly the eigenfrequencies of the system. In contrast, with the mode shape isn't possible to distinguish between hunting and non-hunting scenarios due to the scaling of the mode shapes themselves. To solve the problem, the following equality is assumed,

$$R = \Phi b, \quad (6)$$

where b is the scaling factor and R is the signals $n \times 1$ RMS matrix. In this way, it is possible to incorporate the energy information carried by the whole considered signal in the mode shapes themselves. In Figure 2 the effect is shown for the first 25 test cases. After the scaling, the mode shapes can be used in conjunction with eigenfrequencies to distinguish hunting cases from non-hunting ones using a statistical fault classifier.

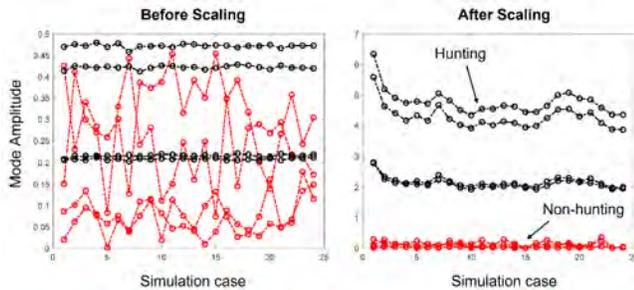


Figure 2 Effect of DMD mode shape scaling with signal RMS

2.3. Intelligent Fault Detection of Vehicle Running Instability with TCN

2.3.1. Estimation of Transfer Function

A rail vehicle running on track in presence of track irregularities can be considered a MIMO system, where Alignment Level (AL), Longitudinal Level (LL), Track Gauge (TG), and Cross Level (CL) are four input signals and vehicle accelerations in X, Y, and Z directions (i.e. longitudinal, lateral and vertical direction) are output signals. Thus, the transfer functions and coherence between carbody floor accelerations and track irregularities are estimated according to principals of MIMO system identification. The schematic of the MIMO system is shown in Figure 3 The simplified relationship between the input and output signal is modelled by linear, time-invariant Transfer Functions.



Figure 3 Modelling a Rail Vehicle as a MIMO system (A simplified schematic)

2.3.2. Transfer Function Estimation Case Study

In this case study, the EUROFIMA coach (Iwnicki, 1999) is running on a 2 km tangent track section in presence of AL, LL, TG, and CL irregularities. These track geometry irregularities are distributed among classes A, B and C defined in the European Standard *EN 13848-5:2017* (2017) standard and the irregularity signals are free from defects. The track irregularities are shown in Figure 4 (a-d). For simulations three different wheel-rail conditions are

considered, see Table 1. In the three cases, a worn wheel profile (T19) is applied to all wheels of the coach. Two rail profiles, namely MS3_MS4 (ground rail) and BDL354U28 (worn rail) are applied to the rails. Case1: No fault at the wheel-rail interface; Case2: Tight gauge fault; Case3: worn rail profile fault (see Table 1). Running equivalent conicity for the three simulation cases is shown in Figure 3(e). Case1: low conicity conditions; Case2: high conicity conditions caused by tight track gauge; Case3: high conicity conditions caused by worn rail profile.

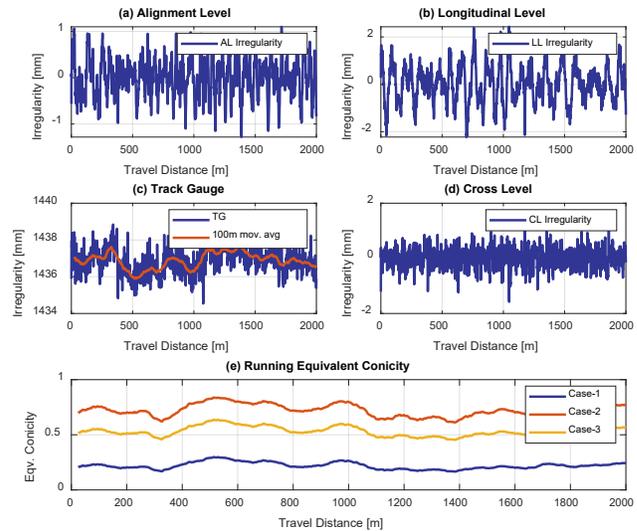


Figure 4 Track irregularities (a-d) and running equivalent conicity (e)

Table 1 Summary of Simulation Cases

Case Number	Wheel Profile	Rail Profile	Avg TG Range [mm]
Case 1	T19	MS3_MS4	1436-1438
Case 2	T19	MS3_MS4	1432-1434
Case 3	T19	BDL354U28	1436-1438

The simulated vehicle response at the carbody floor (above the left axlebox of the leading wheelset) is stored in GENSYS. The differences between the three cases is shown by the X and Y accelerations of the carbody floor and EN 14363 stability evaluation (i.e., 100 m moving RMS of bandpass filtered lateral bogie frame acceleration) for the three simulated cases are shown in Figure 5. In each subfigure abscissa and ordinate axes are travel distance and acceleration, respectively. In case1, X&Y acceleration amplitudes are low (see subfigure a, b) and the lateral bogie frame acceleration is much lower than the limit value (subfigure c). In case2, the vibration level is very large, especially the lateral acceleration (see subfigures d, e). The lateral bogie frame acceleration exceeds the threshold value

on many occasions on this 2 km section as seen in subfigure (f). In case3, also the vibration level is strong and to the order of 0.8 m/s² (see subfigures g, h). The lateral bogie frame acceleration is high but always lower than the threshold value throughout this 2 km section as seen in subfigure (i).

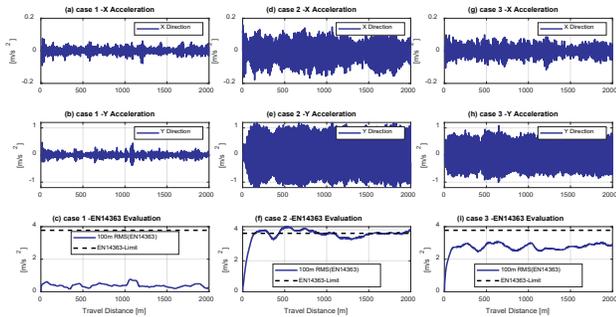


Figure 5 X, Y raw carbody floor accelerations and stability evaluation of bogie frame according to European Standard EN14363 scheme in case1, case2, and case3.

The lateral carbody floor acceleration is processed through the first feature extraction algorithm to obtain transfer functions. The transfer functions for Y & AL and Y & CL are shown in Figure 6. The functions are presented as distance–frequency plots to obtain spatial and frequency localization of the vehicle behaviour. In each plot, abscissa and ordinate are travel distance and frequency, respectively, whereas the colour shows the transfer function’s magnitude in dB scale. In case1, the magnitude of both transfer functions is always below 0 dB throughout the travel distance (see subfigures a, b). In case2, the Y vs AL transfer function (subfigure c) peaks in the 5-6 Hz range with amplitude above 30 dB and amplitude below 0 dB elsewhere. The magnitude of the Y vs AL transfer function does not change much throughout travel distance except for 200 m around the 1000 m marker. Similarly, the Y vs CL transfer function (subfigure d) of case2 exhibits peaks at 5-6 Hz with amplitude in the 0-10 dB range. In case3, the Y vs AL transfer function (subfigure e) peaks in the 5-6 Hz range with an amplitude of 10-20 dB throughout the travel distance. Similarly, the Y vs CL transfer function (subfigure f) of case3 does not exhibit strong peaks at 4-6 Hz.

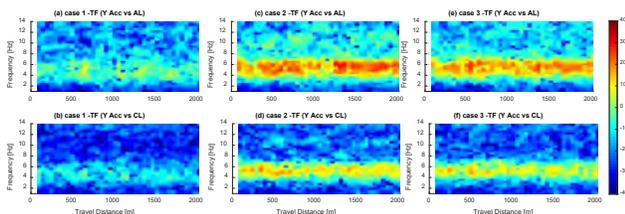


Figure 6 Transfer Functions between Y acceleration and AL & CL track irregularities for case1, case2 and case3

The magnitude of the transfer function between carbody floor and track at a particular frequency varies with time as the

vehicle travels on track mainly because of variations in equivalent conicity. This varying magnitude of the transfer function at a specific frequency is time-series data and the transfer function contour plots shown above are a collection of time series which are highly nonlinear. This time-series data is used for the identification of root causes of observed vehicle running instability.

2.3.3. Temporal Convolutional Network (TCN)

A Convolution Neural Network (CNN) is a classical neural network that performs well at image processing tasks because of its excellent feature extraction capabilities. Currently, CNN has been widely used in many fields such as face recognition, automatic driving, and security. However, CNN models are poor in extracting temporal features from the data. In recent times, a Temporal Convolutional Network (TCN) is proposed which shows excellent abilities in solving sequential problems such as analyzing time series data and outperforms Recurrent Neural Network (RNN) models. Thus, TCN is deployed to identify the root causes of observed vehicle running instability in this investigation.

Generally speaking, TCN has two main characteristics. Firstly, it maintains a causal relationship between each layer of the network, which means that the convolution output of a layer is determined solely on the convolution result of layers before. Thus, the data coherence and time coherence are better protected than the limited historical information storage and possible data absence of LSTM’s memory cell. Secondly, the architecture of this model can be flexibly adjusted to any length. It can also be mapped according to several interfaces required by the output, which is similar to the RNN framework. Compared with the traditional CNN network structure, TCN adds four core parts to the design: sequence modelling, causal convolutions, dilated convolutions, and residual connections. This subsection will introduce the architecture and working principle through these four parts in brief.

1. Sequence Modelling: A simple sequence modelling task is used to illustrate the sequence modelling characteristics of TCN. If the input sequence is given, it requires predicting the specific outputs O_0, \dots, O_T at every step. Following the requirements, the model should predict the corresponding output at a particular time point. The key constraint of sequence modelling is that the output at a time should be generated by exactly the recorded inputs before time t instead of the post-positional information, which follows the sequence of data flow. The one-to-one mapping from it to y_t of sequence modelling network could be simply expressed as:

$$\hat{O}_0, \dots, \hat{O}_T = f(i_0, \dots, i_T) \quad (7)$$

- Causal Convolutions:** After the introduction of the sequence modelling above, two principles of TCN are summarized. First, the length of output after model prediction will always remain the same as the input length. Second, the TCN remains invisible to ‘future’ information and always depends on the previous inputs to complete the prediction. To maintain the first principle, the TCN utilizes the 1D fully-convolutional network (FCN). The core idea of FCN is adopting the zero-padding method to guarantee that each output layer keeps the same length and width as the input layer in the propagation of the network. As for the second principle, TCN utilizes causal convolutions to prevent future information leakage. Causal convolutions are abstracted to predict current output y_T depending on previous inputs x_0, \dots, x_T and previous layers’ output y_0, \dots, y_{T-1} to approach the actual value. The example of causal convolution is shown in Figure 7

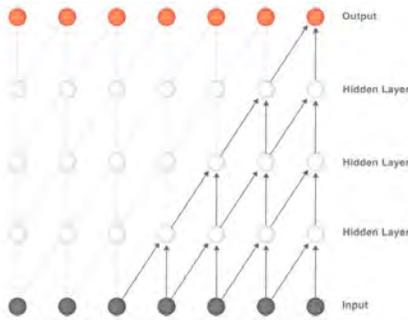


Figure 7 An example of causal convolutions

- Dilated Convolutions:** Although the above causal convolutional structure is feasible to prevent future information leakage, it increases the number of layers in the network and keeps extremely long historical information sequences simultaneously. As Figure 7 shows, the signed output in the upper right corresponds to five perceptive fields (5 grey balls in the input sequence), and it is obtained through five layers. It shows that the size of the receptive field has a positive linear correlation with the depth of the network, which may burden the learning process. To simplify the network and relieve memory storage pressure, TCN applies dilated convolutions on the network and forms an exponential correlation between the size of the receptive field and the number of layers. The following equation can demonstrate the principle:

$$F(s) = (x *_{d} f)(s) = \sum_{i=0}^{k-1} p(i) \cdot x_{s-d \cdot i} \quad (8)$$

Where d is the dilation factor and k is the filter size which $s - d \cdot i$ means convoluting only the former state. x is the sequence input and $f : \{0, \dots, k - 1\}$ is the filter. The

operation F takes the input s to complete convolutions using a fixed step between every two adjacent filter taps. Figure 8 shows the different dilated convolutions when d is 1, 2, and 4 respectively, the whole architecture of the network becomes dilated and includes less historical data. Therefore, this method can keep a large perceptive field with fewer layers and simplify learning tasks.

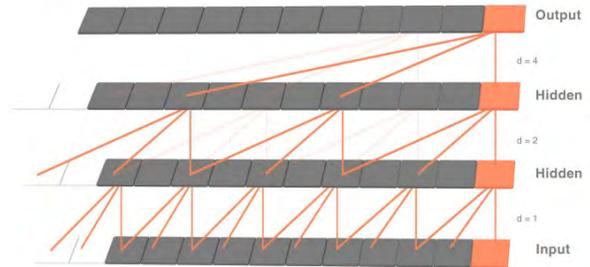


Figure 8 An example of dilated convolutions.

- Residual Connections:** The fast track in ResNet enables the model to learn the difference information, which effectively allows the network to modify the identity mapping to avoid gradient vanishing and gradient exploding problems in the deep layer model. For TCN, if the model needs to record a large amount of historical information, the final receptive field could be vast, and the network could become extremely deep. Hence, TCN adopted residual connections to reduce network depth. Each residual block module consists of two layers of residual convolutions, ReLU and batch normalization operation. In addition, spatial dropout is added after the activation function. An illustration of detailed residual block construction is in Figure 9.

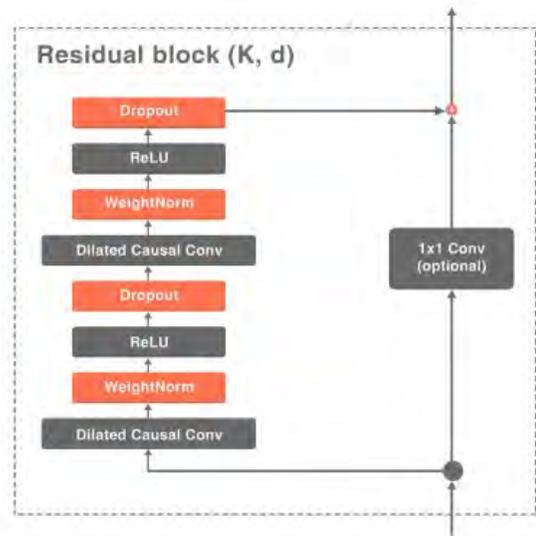


Figure 9 The profile of one residual block in TCN.

The advantages of TCN are -

1. TCN can conduct convolution operations in parallel. Therefore, TCN can preserve long-term memory in both training and validation.
2. Gradient stable TCN has a different backpropagation path from the sequence time direction, which avoids the gradient exploding and gradient vanishing problems in deep-layer networks compared to the RNN.
3. The TCN can possess a sizeable perceptive field under the condition of shallow layers. Therefore, TCN can be more flexible in the model’s memory size, and it is easy to migrate to other fields.
4. The TCN can accept any length of the input sequence by sliding one-dimensional convolutional kernels. Therefore, it is flexible to be utilized on distinct tasks.

The disadvantages associated with TCN are -

1. To maintain the long-term memory and generate the predicted result, the TCN needs to occupy more memories during the testing phase.
2. When TCN migrates to different fields, the requirement of historical length and perceptive field will be distinct. Hence, migration operations could result in a weak expression of the TCN model.

3. FORMULATION OF VEHICLE RESPONSE DATABASE

3.1. Vehicle Model

In this investigation, the hunting behaviour of a vehicle is investigated using the commercial multibody dynamics software GENSYS (AB DEsolver) by performing time domain simulations. The Swedish train operator SJ operates the fast trains X2000 on the Swedish rail network. Most SJ X2000 trains consist of a power car, five intermediate coaches and a driving trailer and are operating at a top speed of 200 km/h. An intermediate coach is modelled here in GENSYS. The vehicle model consists of a carbody, two bogie frames and four wheelsets which are modelled as 6 DOF rigid bodies and connected by primary and secondary suspension elements. The primary and secondary suspensions consist of spring and viscous damper elements in the x, y, and z-directions. Since the X2000 coach is specifically designed to run in curves at high cant deficiencies, the primary suspension is relatively soft to give the wheelsets improved radial self-steering capabilities. The X2000 coach model is also equipped with four yaw dampers as shown in Figure 10 i.e., two per bogie, which works in the longitudinal direction.

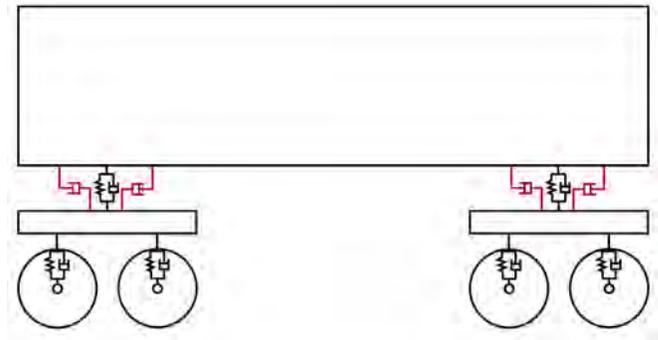


Figure 10 Schematic of MBS model of rail vehicle (Side view)

3.2. Vehicle Dynamic Simulations

Vehicle running instability can be caused by various parameters such as poor conditions of track gauge, suspension components and wheel-rail interfaces. In this investigation, the simulations are carried out with variation of wheel-rail friction, equivalent conicity and yaw damper as these factors mainly affect the running stability. Therefore, 384 simulation cases were performed with the combination of 3 friction values, 8 conicity cases and 2 damping coefficients for each yaw damper as summarized in Table 2. In total 384 cases are obtained with a full factorial design of the 6 parameters.

Table 2: Simulation Parameters

Parameter	Values
Friction	0.1, 0.35, 0.6
Conicity	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8
Damping coefficient of yaw damper	10% and 100% of the designed value

The vehicle dynamic responses are measured with two accelerometers at two distinct diagonal locations on the carbody floor as illustrated in Figure 11. The data obtained from these simulations are used for implementing the proposed iVRIDA algorithm.

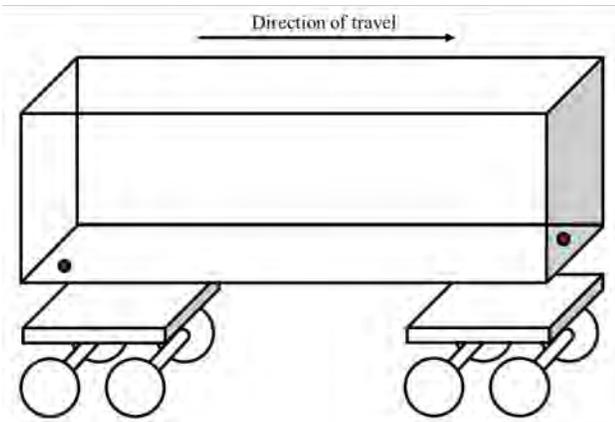


Figure 11 Sensor locations for acceleration measurements

3.3. Machine Learning problem formulation

Vehicle Running Instability Detection with DMD

The 3 features obtained from the DMD analysis of 384 cases, namely frequency and normalized mode shapes (at two sensor locations) are used for training and testing the fault classifier. The true labels are generated using the running instability evaluation scheme defined in EN14363. The EN 14363 scheme is typically used in the railway industry to classify the running state of the vehicle as stable or unstable, thus this is a typical binary classification problem, and any typical statistical classifier can perform the above-mentioned classification task. Thus, in this investigation, Linear SVM is deployed. The database of 384 cases is divided with random stratification into training and testing datasets with 87.5% and 12.5% cases respectively. The Linear SVM (L-SVM) classifier is trained on a training dataset with 7-fold crossvalidation and the hyperparameters of the L-SVM are optimized. The results are presented in the result section.

Intelligent Fault Detection of Vehicle Running Instability with TCN

The time-series form of transfer functions as explained in 2.3.2 of each case are used for the identification of root causes of observed vehicle running instability using TCN. The time series of all 384 cases are horizontally stacked together to obtain a very large matrix. This large collection of time series of 384 cases is used for training and testing the TCN. The true labels corresponding to each case are obtained from the simulation parameters. The simulations with a 10% damping coefficient of the yaw dampers are labelled as yaw damper faults. There are 16 classes of yaw damper fault conditions as there are four yaw dampers in the vehicle and one or more may fail simultaneously. Thus, this is a typical multiclass classification problem. The database of 384 cases is randomly divided with stratification into training and testing datasets with 87.5% and 12.5% cases respectively. The training dataset is stratificaly divided into 7 folds and the first 6 folds formed 6 batches for the training of TCN. The last 7th fold is

used as the validation set and the best performing TCN on the validation dataset is tested on the test dataset.

4. RESULTS

4.1. Vehicle Running Instability Detection with DMD

The DMD algorithm is applied to the simulation cases described in subsection 3.2. The results are shown in **Figure 12**, subfigure a & b corresponds to instable cases and subfigures c & d to stable cases. Subfigures a & c shows the carbody vibration frequency of instable and stable cases whereas subfig b & d shows corresponding normalized mode shapes. It can be seen, two distinct families corresponding to instable and stable cases are now distinguishable. In the mode shape of the hunting cases, it is now possible to distinguish which part of the carbody is most excited, front (111), rear (122) or both.

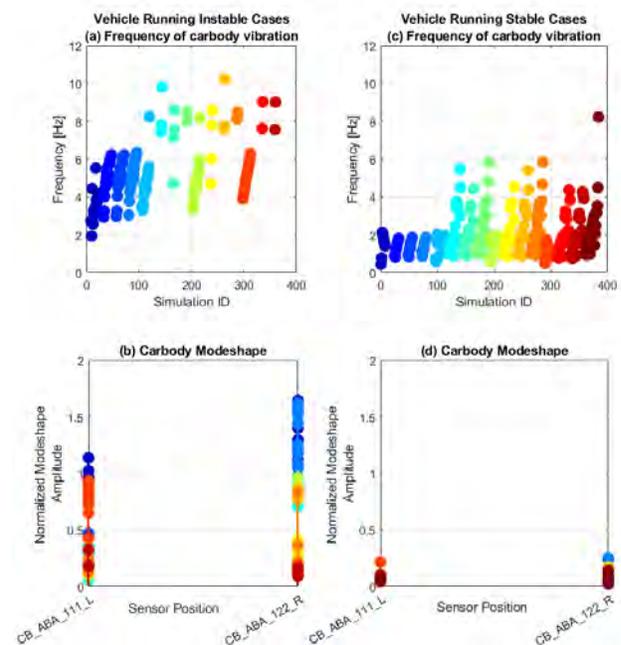


Figure 12 Frequencies and mode shapes for instable and stable cases detected with the DMD algorithm

The performance of the L-SVM classifier on the DMD dataset is shown as a confusion matrix in **Figure 13**, subfigure a & b show the performance of LSVM in the training and testing phase respectively. The LSVM classifies all cases with 100% accuracy in both the training and testing phase.

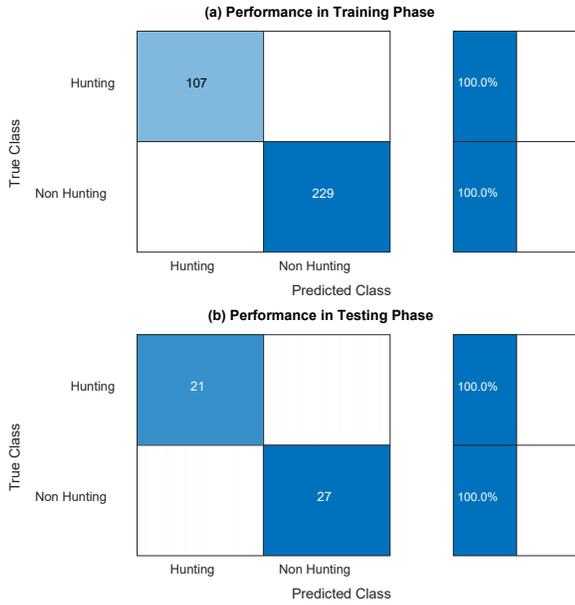


Figure 13 Detection of vehicle running instability with DMD+LSVM

4.2. Intelligent Fault Detection of Vehicle Running Instability with TCN

The performance of TCN in this investigation is evaluated with help of a confusion matrix where true labels are on the y-axis and predicted labels on the x-axis. The row-wise performance is summarised on the right-hand side of the respective confusion matrix. The results obtained during the testing phase are presented in Figure 14, and the trained fault classifier identifies root causes with 98.7% accuracy.

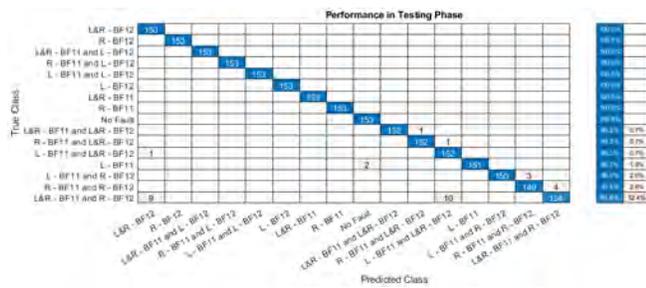


Figure 14 Intelligent Fault Detection of Vehicle Running Instability with TCN

The comparison of predicted fault labels and true fault labels is shown in Figure 15. In the figure, the x-axis is the test observation ID, the y-axis is the fault labels. The true fault labels are shown with a blue line and predicted labels with a black solid circle. In the well-trained fault classifier, ideally, the black solid circles should follow the blue line. It can be

observed in the figure that the TCN fault classifier’s performance is very accurate across the whole test sequence except a few misclassifications.

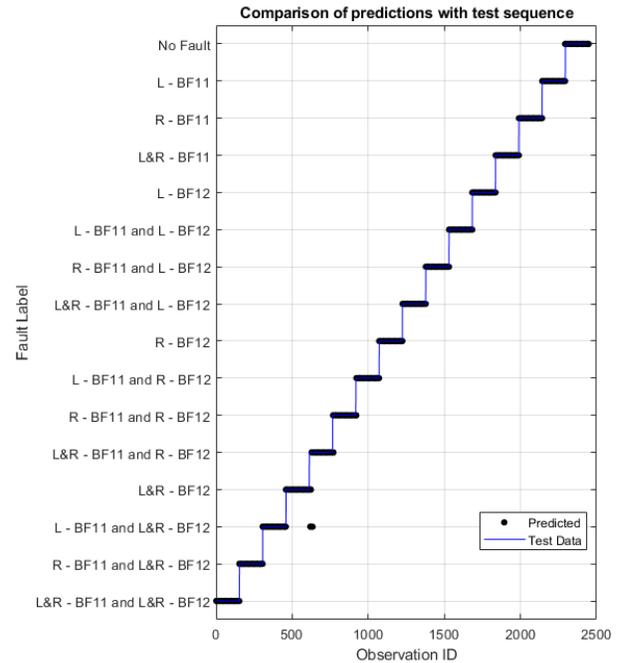


Figure 15 Comparison of predicted labels with true labels in the test sequence

5. CONCLUSIONS

In this paper, a data-driven intelligent vehicle running instability detection method is proposed for detecting and identifying the root cause of vehicle running instability of fast railway vehicles. The proposed novel methodology utilises carbody floor accelerations for intelligently detecting the vehicle faults exciting vehicle running instability. The iVRIDA algorithm detects vehicle running instability with the DMD+SVM method and corresponding root causes with Temporal Convolutional Network (TCN). In this investigation, both fault detection models are trained and tested with an extensive database generated with numerical experiments. The DMD+SVM algorithm detects the stability of high-speed rail vehicles from carbody floor accelerations with 100% accuracy. The TCN based fault classifier identifies the root cause of running instability with 98.7% accuracy. Thus, it is significant that iVRIDA detects and isolates the occurrence of vehicle running instability and corresponding root cause from carbody floor accelerations. The most important benefit of the proposed novel deep learning algorithm is the enhancement in obtaining a reliable Intelligent fault detection method with minimal sensor maintenance.

6. REFERENCES

- AB DESolver. (n.d.). *GENSYS: A software tool for modelling and simulating vehicles running on rails*. <http://gensys.se/>
- Brunton, S. L., & Kutz, J. N. (2019). Data-Driven Science and Engineering. In *Cambridge University Press*. Cambridge University Press. <https://doi.org/10.1017/9781108380690>
- EN 13848-5:2017 - *Railway applications – Track – Track geometry quality – Part 5: Geometric quality levels – Plain line, switches and crossings*. (2017).
- EN 14363:2016+A1 - *Railway applications – Testing and simulation for the acceptance of running characteristics of railway vehicles – Running behaviour and stationary tests*. (2019).
- Gasparetto, L., Alfi, S., & Bruni, S. (2013). Data-driven condition-based monitoring of high-speed railway bogies. *International Journal of Rail Transportation*, 1(1), 42–56. <https://doi.org/10.1080/23248378.2013.790137>
- Iwnicki, S. (1999). The Manchester Benchmarks for Rail Vehicle Simulation. In S. Iwnicki (Ed.), *Vehicle System Dynamics* (Vol. 31, Issue Supplement).
- Kulkarni, R., Qazizadeh, A., Berg, M., & Stichel, S. (2019). Fault Detection and Isolation Method for Vehicle Running Instability from Vehicle Dynamics Response Using Machine Learning. *Proceedings of BOGIE'19*.
- Lea, C., Vidal, R., Reiter, A., & Hager, G. D. (2016). Temporal convolutional networks: A unified approach to action segmentation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 9915 LNCS* (pp. 47–54). https://doi.org/10.1007/978-3-319-49409-8_7
- Ning, J., Liu, Q., Ouyang, H., Chen, C., & Zhang, B. (2018). A multi-sensor fusion framework for detecting small amplitude hunting of high-speed trains. *Journal of Vibration and Control*, 24(17), 3797–3808. <https://doi.org/10.1177/1077546318787945>
- Zeng, Y., Zhang, W., & Song, D. (2020). A new strategy for hunting alarm and stability evaluation for railway vehicles based on nonlinear dynamics analysis. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 234(1), 54–64. <https://doi.org/10.1177/0954409719830177>

Remaining-Useful-Life prognostics for opportunistic grouping of maintenance of landing gear brakes for a fleet of aircraft

Juseong Lee¹, Ingeborg de Pater², Stan Boekweit³ and Mihaela Mitici⁴

^{1,2,3,4} *Delft University of Technology, Faculty of Aerospace Engineering, Delft, 2629HS, the Netherlands*

J.Lee-2@tudelft.nl

I.I.dePater@tudelft.nl

stanboekweit@gmail.com

M.A.Mitici@tudelft.nl

ABSTRACT

Several studies have proposed Remaining-Useful-Life (RUL) prognostics for aircraft components in the last years. However, few studies focus on integrating these RUL prognostics into maintenance planning frameworks. This paper proposes an optimization model for opportunistic maintenance scheduling of aircraft components that integrates RUL prognostics and that groups the maintenance of these components to reduce costs. We illustrate our approach for the maintenance of a fleet of aircraft, each equipped with multiple landing gear brakes. RUL prognostics for the landing gear brakes are obtained using a Bayesian regression model. Based on these RUL prognostics, we group the replacement of brakes using an integer linear program. As a result, we obtain a cost-optimal RUL-driven opportunistic-maintenance schedule for the brakes of a fleet of aircraft. Compared with traditional maintenance strategies, our approach leads to a reduction of up to 20% of the total maintenance costs.

1. INTRODUCTION

Remaining-useful-life (RUL) prognostics are regarded as a key enabler for predictive aircraft maintenance (Sprong, Jiang, & Polinder, 2019). Using RUL prognostics, predictive maintenance aims to perform maintenance tasks in anticipation of failures of aircraft components. The expected impact of predictive maintenance is to reduce unexpected failures, increase system availability, and reduce overall maintenance costs (Lee & Mitici, 2022).

Several studies have proposed algorithms for RUL prognostics for various aircraft systems. For example, Mitici and de Pater develop prognostics for aircraft cooling units using particle filtering. Lee and Mitici propose a regression model to

characterize the degradation of aircraft landing gear brakes. Eleftheroglou et al. present the data-driven prognostics for batteries of unmanned aerial vehicles. de Pater, Reijns, and Mitici predict the RUL of aircraft engines using a convolutional neural network and the C-MAPSS data set (Saxena & Goebel, 2008).

Despite the increasing number of RUL prognostics for aircraft systems, few studies integrate these prognostics into actual maintenance planning frameworks to prescribe RUL-driven maintenance tasks (de Jonge & Scarf, 2020; de Pater & Mitici, 2021; Kim, Choi, & Kim, 2022). Such integration is particularly complex since aircraft maintenance planning should consider, apart from RUL prognostics, additional factors such as the flight schedule, the limited availability of the hangar where aircraft are maintained, the cost of different maintenance tasks, and the management of spare parts (de Pater & Mitici, 2021).

Moreover, when considering multiple components, it is desirable to group maintenance tasks to reduce maintenance setup costs (Wildeman, Dekker, & Smit, 1997; Bouvard, Artus, Bérenguer, & Cocquempot, 2011). The approach of grouping maintenance tasks is referred to as opportunistic maintenance (OM). Several studies have proposed OM for various applications, especially for the maintenance of wind turbines (Vu, Do, Fouladirad, & Grall, 2020; Aizpurua, Catterson, Papadopoulos, Chiacchio, & D'Urso, 2017; Xia et al., 2021). However, existing studies are not readily applicable for predictive maintenance of a fleet of aircraft because they consider neither RUL prognostics (Vu et al., 2020), nor the limited availability of critical resources such as hangars (Aizpurua et al., 2017), nor the fact that the flight schedule of aircraft restricts the planning of maintenance (Xia et al., 2021). Thus, these critical constraints need to be considered for the OM for a fleet of aircraft.

In this paper, we integrate RUL prognostics of aircraft components into opportunistic maintenance (OM) for a fleet of

Juseong Lee et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

aircraft. Our approach groups maintenance tasks for aircraft components based on their RUL prognostics. The goal of grouping the maintenance of the components is to reduce maintenance setup costs, i.e., the costs needed to initiate maintenance. We illustrate our approach for landing gear brakes of a fleet of aircraft. We first propose a Bayesian linear regression model to predict the RUL of aircraft landing gear brakes. The obtained RUL prognostics are validated against sensor measurements obtained during the actual operation of the aircraft. Then, taking into account these RUL prognostics, we propose an integer linear programming model to opportunistically plan maintenance for the brakes. Our model considers the limited availability of hangars where maintenance can be performed, as well as realistic flight schedules. The result shows that the proposed RUL-driven OM reduces by 20% the expected total maintenance cost for the brakes of a fleet of aircraft compared to traditional maintenance approaches.

2. RUL PROGNOSTICS FOR AIRCRAFT LANDING GEAR BRAKES

2.1. Maintenance of aircraft landing gear brakes

We consider the maintenance of landing gear brakes of wide-body aircraft. A wide-body aircraft is equipped with 8 landing gear brakes, 4 on each side of the wings. The carbon disks of the brakes are worn out when the aircraft decelerates. As soon as the remaining thickness of a braking disk is below an operational threshold, it needs to be replaced before the aircraft can perform another flight.

According to current maintenance practice, aircraft landing gear brakes are inspected periodically (Lee & Mitici, 2020). Every d flight cycles, mechanics measure the remaining thickness of the brakes. If the remaining thickness is below a predefined threshold, then the brake is replaced with a new one. In order to ensure a high reliability, the inspection interval d is often short, i.e., frequent inspections. Using RUL prognostics, predictive maintenance aims to reduce the wasted life of brakes due to too-early replacements, while limiting the cases when the degradation of a brake may unexpectedly exceed an operational threshold.

2.2. Condition monitoring of aircraft landing gear brakes

New aircraft are equipped with brake condition monitoring systems that measure the thickness of the brake disks. The thickness of a disk is a direct measure of the degradation level of a brake. Formally, let us denote the degradation level of a brake after ϕ^{th} flight cycle as Z_ϕ . We normalize this degradation level so that $Z_\phi = 0$ when the brake is new. As soon as $Z_\phi > \eta$, where $\eta = 1$ following normalization, the brake needs to be replaced. As soon as $Z_\phi > \eta$, we say that the brake becomes *inoperable*.

In this study, we analyze the actual brake degradation data

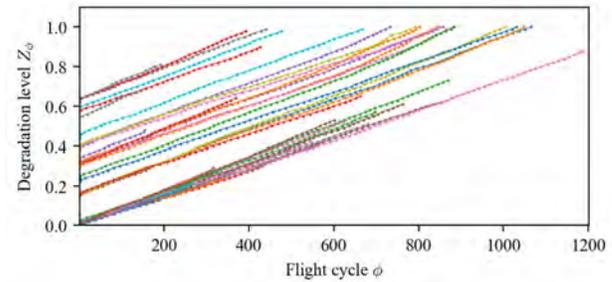


Figure 1. The degradation data of landing gear brakes.

collected from a fleet of aircraft. These aircraft have been in operation for a period of 6 months up to 3 years. Figure 1 shows the normalized degradation data recorded for several aircraft. The x -axis is the number of flight cycles (ϕ) during which a brake was used, and the y -axis is the degradation level (Z_ϕ) of the brakes. The line segments of different colors represent different brakes. Figure 1 shows that the degradation of a brake continuously and stochastically increases over time.

Under predictive maintenance, the goal is to use the information provided by RUL prognostics to replace brakes just before their degradation reaches an operational threshold ($\eta = 1$). In Figure 1, the end of a line segment is the moment when the brake is replaced under the current practice. We note that in current practice, RUL prognostics are not yet utilized to plan maintenance. Often, brakes are preventively replaced before their degradation level reaches threshold η , wasting the useful life of the brakes. Using RUL prognostics, the aim is to achieve a higher utilization of the brakes while minimizing maintenance costs.

2.3. RUL prognostics of aircraft landing gear brakes

Given the brake degradation data recorded for a fleet of aircraft, we use a Bayesian linear regression (BLR) to predict the remaining-useful-life (RUL) of the brakes. For the brake degradation data in Figure 1, its linearity allows the BLR model to achieve accurate RUL predictions compared to advanced non-linear models such as artificial neural networks (Oikonomou, Eleftheroglou, Freeman, Loutas, & Zarouchas, 2022). The input of the BLR model is the number of flight cycles ϕ , and the output is the (predicted) degradation level of a brake after this flight cycle \hat{Z}_ϕ . Formally, we consider the following probabilistic model:

$$P(\hat{Z}_\phi | \phi, \omega, \sigma) = \mathcal{N}(\hat{Z}_\phi | \phi\omega, \sigma^2), \quad (1)$$

where ω is the coefficient of the linear model, and σ^2 is the variance of the Gaussian model. The prior of the coefficient ω is assumed to be zero-mean Gaussian, i.e., $P(\omega) = \mathcal{N}(\omega | 0, \lambda\mathbf{I})$. Here, λ and σ^2 are the hyper-parameters of the model, and we consider a Gamma distribution as their prior. Finally, the pa-

rameters ω , λ , and σ^2 are jointly optimized by maximizing the log marginal likelihood (Pedregosa et al., 2011).

Then, given that a brake is already operated for ϕ flight cycles, its RUL $\rho(\phi)$ is the number of remaining flight cycles until the probability that the degradation level exceeds η is larger than a threshold ξ , i.e.,

$$\rho(\phi) = \min_{\delta} \left\{ \delta : P(\hat{Z}_{\phi+\delta} \geq \eta | Z_{\phi}) \geq \xi \right\}, \quad (2)$$

where ξ is a given reliability threshold.

The RUL prognostics of the brakes are updated after every flight cycle, taking into account the most recently available degradation data collected from the on-board condition monitoring systems.

A result of RUL prognostics of a brake in the actual data set is shown in Figure 2. We predict the RUL of this brake after it has been operated for 748 flight cycles. Given the degradation, the degradation level is expected to exceed $\eta = 1$ after 40 flight cycles with probability $\xi = 0.5$, and thus, the predicted RUL is $\rho(\phi) = 40$. Given the true RUL $\rho^* = 44$, the error of the RUL prediction is -4 flight cycles.

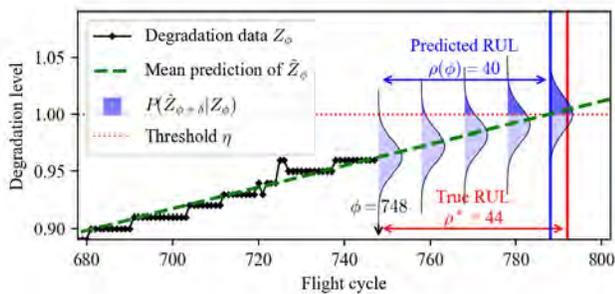


Figure 2. Result of RUL prognostics obtained for a brake in the data set. The predicted RUL is 40 cycles and true RUL is 44 cycles.

2.4. Performance of the RUL prognostics

The performance of the proposed RUL prognostics using BLR is validated based on the actual degradation data collected from a fleet of aircraft. We consider the sensor measurements of 40 brakes of a fleet of aircraft which have been operated in real-life conditions. Each of these 40 brakes have been operated for ϕ^* flight cycles until these brakes become inoperable, i.e., $Z_{\phi^*} = \eta$. Their recorded degradation data are used as a test set for our BLR model since we know the true RUL of the 40 brakes.

We apply BLR at several moments during the operation of the brakes: at 200, 100, 50, and 25 flight cycles before the brakes become inoperable, i.e., the true RUL at these moments in time is $\rho^* \in \{200, 100, 50, 25\}$ flight cycles. We

predict the RUL of 40 test brakes at these moments, and plot the box plots of the error $\rho - \rho^*$ in Figure 3. We also determine the mean-bias-error (MBE) and root-mean-squared-error (RMSE) as follows:

$$MBE = \frac{1}{K} \sum_{k=1}^K (\rho_k - \rho_k^*),$$

$$RMSE = \frac{1}{K} \sqrt{\sum_{k=1}^K (\rho_k - \rho_k^*)^2},$$

where $K = 40$ brakes considered. Table 1 shows the MBE and RMSE of the proposed RUL prognostics.

The error of the RUL prognostics is smaller when true RUL is smaller, i.e., the accuracy of the prognostics increases as we approach the time of failure. In particular, MBE is smaller than 2 flight cycles when the true RUL is 100 flight cycles (see Table 1). Considering the fact that an aircraft makes 2 flights per day on average, the bias of the prognostics is roughly 1 day only. Moreover, the RMSE decreases to 5.4 flight cycles, which is very small considering the average useful life of the brakes in our model (approximately 1250-1450 flight cycles) (Lee & Mitici, 2022). Based on this performance of the BLR, we conclude that our prognostics are reliable to be used for maintenance scheduling.

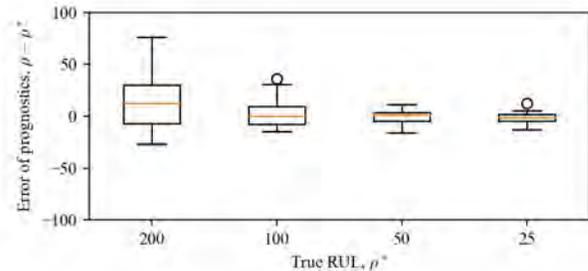


Figure 3. Error of the RUL prognostics for the brakes in the data set.

Table 1. Performance of the proposed RUL prognostics for the brakes in the data set.

True RUL ρ^* [Flight cycles]	200	100	50	25
MBE [Flight cycles]	8.4	1.7	-0.5	-1.7
RMSE [Flight cycles]	41.3	12.4	6.0	5.4

3. INTEGRATION OF RUL PROGNOSTICS INTO OPPORTUNISTIC MAINTENANCE SCHEDULING

We propose a RUL-driven opportunistic maintenance planning (RUL-driven OM) for a set of generic aircraft components whose degradation is monitored over time and whose RUL is updated over time. We propose an integer linear pro-

gramming (ILP) model to group maintenance tasks for these components considering their RUL prognostics.

3.1. Problem description

Our goal is to schedule the maintenance of multiple components of a fleet of aircraft, while minimizing the total maintenance cost. We consider M aircraft ($i \in \mathcal{I} = \{1, \dots, M\}$), each equipped with N components ($j \in \mathcal{J} = \{1, \dots, N\}$). The aircraft perform a sequence of flights according to a flight schedule. Figure 4 shows an example of a historical flight schedule. The components are used during flight-time when their degradation evolves stochastically over time. Based on the flight schedule, we define maintenance slots, which are time periods when the aircraft is on-ground at an airport with a hangar. The aircraft can undergo maintenance only at the hangar. Due to the limited space and resources at the hangar, at most H aircraft can be maintained at the same time in the hangar.

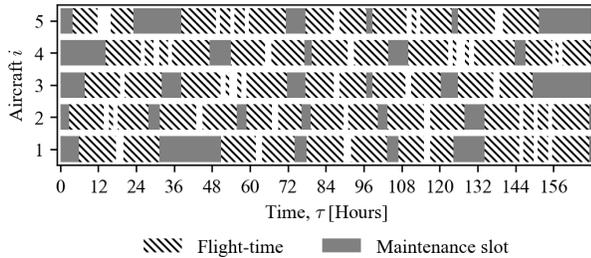


Figure 4. An example of flight schedules for 5 aircraft for a week.

The cost of aircraft maintenance consists of i) the setup cost and ii) the component replacement cost. The setup cost C_{set} is the cost to prepare the maintenance of an aircraft in the hangar. This cost can be reduced if multiple maintenance tasks are grouped and performed together during one hangar visit.

Over time, components are scheduled for replacement several weeks in advance. The cost of a scheduled replacement for a component is C_{sch} . If, however, this component becomes inoperable unexpectedly before the moment of the scheduled replacement, we perform an unscheduled replacement for this component at cost C_{uns} . In general, we assume $C_{\text{uns}} > C_{\text{sch}}$ (Pereira, Gomes, Melicio, & Mendes, 2021).

3.2. Rolling horizon for RUL-driven OM

We consider a sequence of time windows that move forward, using a rolling horizon approach (see Figure 5). The r^{th} time window is the time period $[T_0^r, T_1^r]$. At the beginning of each time window, we update the RUL prognostics using the most recent degradation data collected until $\tau < T_0^r$. In addition, we know the maintenance slots available for the fleet

of aircraft during this time window, and the availability of the hangar H . Taking into account this information, we optimize the maintenance schedule for the time window $[T_0^r, T_1^r]$ (see Section 3.3).

Having obtained a maintenance schedule for time window $[T_0^r, T_1^r]$, we roll forward Δ days. The maintenance schedule for the time period $[T_0^r, T_0^{r+1}]$ is fixed, $T_0^{r+1} = T_0^r + \Delta$. If during $[T_0^r, T_0^{r+1}]$ a component becomes inoperable before its scheduled maintenance, then we perform unscheduled maintenance. We next optimize the maintenance schedule for the new time window $[T_0^{r+1}, T_1^{r+1}]$, updating the RUL prognostics.

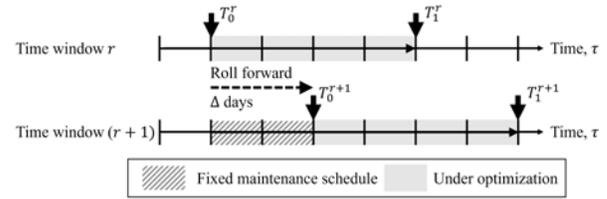


Figure 5. Rolling horizon approach.

3.3. Integer Linear Programming of RUL-driven OM

3.3.1. Decision variables

We define the following two decision variables $x_{i,j,t}$ and $y_{i,t}$:

$$x_{i,j,t} = \begin{cases} 1 & \text{if component } j \text{ of aircraft } i \text{ is scheduled} \\ & \text{for maintenance at time slot } t \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$y_{i,t} = \begin{cases} 1 & \text{if aircraft } i \text{ is scheduled for maintenance} \\ & \text{at time slot } t \text{ but not at time slot } (t-1) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Here, $x_{i,j,t}$ is a binary variable indicating the maintenance schedule, and $y_{i,t}$ is a binary variable indicating the hangar visit of an aircraft. If an aircraft is scheduled for the maintenance of more than 2 components in consecutive time slots, we regard this as one hangar visit, which requires the setup cost once. Thus, $\sum_{t \in \mathcal{T}_i} y_{i,t}$ is the number of hangar visits of aircraft i .

3.3.2. Objective function

We consider the following objective function:

$$\min \sum_{i \in \mathcal{I}} \sum_{t \in \mathcal{T}_i} \left(C_{\text{set}} y_{i,t} + \sum_{j \in \mathcal{J}_i} C_{\text{sch}} x_{i,j,t} + \sum_{j \in \mathcal{J}_i} c_{i,j,t} x_{i,j,t} \right), \quad (5)$$

where the first term is the setup cost for hangar visits, and the second term is the cost for scheduled replacements.

The third term of the objective in Eq. (5) penalizes component replacements that are scheduled too early or too late relative to its predicted RUL. Specifically, the penalty $c_{i,j,t}$ is defined as follows:

$$c_{i,j,t} = \begin{cases} c_1 t - c_2 \rho_{i,j,t} & 0 \geq \rho_{i,j,t} \\ c_3 t & 0 < \rho_{i,j,t} \end{cases} \quad (6)$$

Here, $\rho_{i,j,t}$ is the estimated RUL of component j of aircraft i at time slot t . This RUL is estimated using the prognostics model introduced in Section 2. Also, we assume that $0 < c_1 < c_2 < c_3$.

An example of a penalty $c_{i,j,t}$ in Eq. (6) is shown in Figure 6. If time slot t is before the moment when component j is expected to become inoperable, i.e., if $\rho_{i,j,t} \geq 0$, then the penalty decreases after each flight cycle. Thus, this penalty incentivizes solutions that schedule replacements when RUL is small, i.e., small wasted useful life. When two time slots t_1 and t_2 have the same RUL ($\rho_{i,j,t_1} = \rho_{i,j,t_2}$), the first term in Eq. (6), $c_1 t$, leads to lower penalties for a replacement scheduled at an earlier time slot. On the other hand, if time slot t is after the moment when component j is expected to become inoperable, i.e., if $\rho_{i,j,t} < 0$, then the penalty rapidly increases by $c_3 t$. Thus, with this RUL related penalty $c_{i,j,t}$, our model avoids scheduling a component replacement at a later time than its predicted RUL.

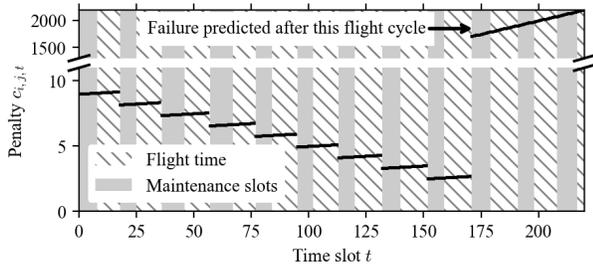


Figure 6. An example of penalty parameter $c_{i,j,t}$ in Eq. (6).

3.3.3. Constraints

The following constraints are considered:

$$\sum_{t \in \mathcal{T}_i} x_{i,j,t} = 1 \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}_i, \quad (7)$$

$$\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}_i} x_{i,j,t} \leq H \quad \forall t \in \mathcal{T}_i, \quad (8)$$

$$\sum_{j \in \mathcal{J}_i} x_{i,j,t} \leq 1 \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T}_i \quad (9)$$

$$\sum_{j \in \mathcal{J}_i} x_{i,j,t} = 0 \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T} : t \notin \mathcal{T}_i, \quad (10)$$

$$\sum_{j \in \mathcal{J}_i} x_{i,j,t} - \sum_{j \in \mathcal{J}_i} x_{i,j,(t-1)} \leq y_{i,t} \quad \forall i \in \mathcal{I}. \quad (11)$$

Constraint (7) ensures all components whose RUL is within the time horizon ($j \in \mathcal{J}_i$) are scheduled for replacements exactly once. Constraint (8) ensures that no more than H aircraft are maintained in the hangar at the same time. In addition, constraint (9) ensures that only one component of an aircraft can be maintained during a time slot t . This constraint (9) is necessary only if $H > 1$. If $H = 1$, then constraint (8) is sufficient. Constraint (10) prevents scheduling maintenance outside of available maintenance slots ($t \notin \mathcal{T}_i$).

Lastly, constraint (11) ensures that the variable $y_{i,t}$ satisfies its definition given in Eq. (4). In particular, constraint (11) provides a lower bound of $y_{i,t}$. So, $y_{i,t} \geq 1$ if the aircraft is brought to the hangar at time slot t , i.e., it is scheduled for maintenance at time slot t , but not at time slot $(t-1)$. On the other hand, $y_{i,t} \geq 0$ if the aircraft is at the hangar at both time slots t and $(t-1)$, or if the aircraft is not at the hangar at both time slots t and $(t-1)$. Since we are minimizing the objective and $C_{\text{set}} > 0$ (see the objective in Eq. (5)), the optimal value of $y_{i,t}$ is its lower bound.

4. NUMERICAL RESULTS:

INTEGRATION OF RUL INTO OM STRATEGY OF AIRCRAFT LANDING GEAR BRAKES

4.1. RUL-driven OM strategy of landing gear brakes

The proposed RUL-driven OM is applied to the maintenance of aircraft landing gear brakes. A wide-body aircraft has 8 brakes ($N = 8$), We consider a fleet of 10 wide-body aircraft ($M = 10$), and assume that at most 1 aircraft can be maintained in a hangar ($H = 1$) at the same time. Using the rolling horizon approach (see Section 3.2), we simulate 10 years of maintenance. The actual degradation of the brakes is shown to follow a Gamma process whose parameters have been estimated in (Lee & Mitici, 2020; van Noortwijk, 2009).

An example of a maintenance schedule generated by our proposed RUL-driven OM is shown in Figure 7. We predict the RUL of components every 2 weeks (the grey vertical lines). The short black vertical lines indicate the moment when the RUL is predicted, the triangles indicate the moment when the component is expected to become inoperable (see Eq. (2)), and the horizontal line segments indicate the length of RUL. Squares indicate the scheduled time of replacements. The optimal solution always allocates the aircraft to maintenance slots within the predicted RUL, i.e., squares are always on the horizontal line segments. The vertical red lines indicate the grouped maintenance tasks. For example, aircraft 1 replaces 6 components with only 3 hangar visits due to grouping: components 5 and 3, components 6 and 2, and components 8 and 4 are grouped together for maintenance. For aircraft 3, component 2 is replaced strictly at RUL without grouping because the closest group of tasks scheduled in November is too early for it, i.e., the benefit of grouping is small.

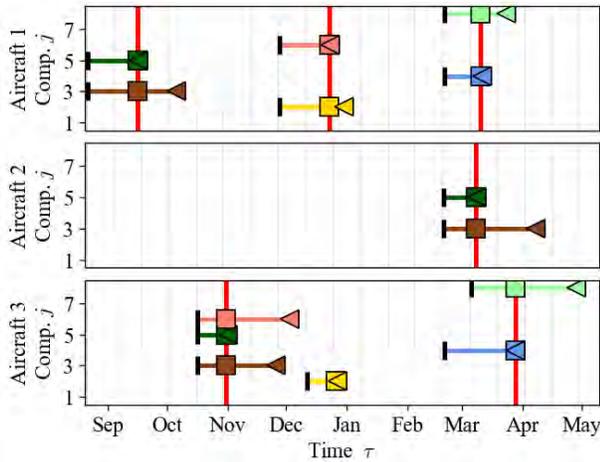


Figure 7. An example of optimal maintenance schedule generated by the proposed RUL-driven OM.

4.2. Benchmarks: traditional maintenance strategies

The performance of the proposed RUL-driven OM is compared with respect to 3 traditional maintenance strategies shown in Table 2. Their schedules are optimized using the ILP in Section 3.3 with modified objective functions, as follows.

1) Preventive maintenance (PM)

Under preventive maintenance (PM), the brakes are replaced at fixed time interval, without making use of the updated condition data or RUL prognostics. Thus, the PM schedule is obtained by modifying the penalty parameter $c_{i,j,t}$ in Eq. (6) as follows:

$$c_{i,j,t} = \begin{cases} c_1 t - c_2 (d_{i,j} - \phi_{i,t}) & \phi_{i,t} \leq d_{i,j} \\ c_3 t & \phi_{i,t} > d_{i,j} \end{cases} \quad (12)$$

Here, $d_{i,j}$ is the deadline to replace brake j of aircraft i , and it is assumed to be the mean-cycles-to-failure of the brakes estimated in (Lee & Mitici, 2020). Also, we set $C_{set} = 0$ in the objective function in Eq. (5) since the setup cost is not considered under PM.

2) Opportunistic maintenance (OM)

Opportunistic maintenance (OM) also replaces components at fixed time intervals, but it does consider the grouping of maintenance tasks to minimize the setup cost. Thus, for OM,

Table 2. Comparison of benchmark strategies.

Strategy	Replacement based on	Considering hangar setup cost
PM	Fixed-interval	No
OM	Fixed-interval	Yes
RUL-driven M	RUL-prognostics	No
RUL-driven OM	RUL-prognostics	Yes

we consider a nonzero C_{set} in the objective function in Eq. (5), and the penalty parameter $c_{i,j,t}$ defined in Eq. (12).

3) RUL-driven maintenance (M)

RUL-driven maintenance (M) schedules all component replacements at the predicted RUL, but without grouping these components. The objective function of RUL-driven M has the same penalty parameter $c_{i,j,t}$ defined in Eq. (6). However, grouping is not performed as setup cost at hangar is not considered, i.e., $C_{set} = 0$.

4.3. RUL-driven OM vs benchmark maintenance strategies

We perform Monte Carlo simulation to evaluate the expected long-run cost of the maintenance strategies in Table 2. The long-run cost is defined as:

$$C = C_{set} N_{hv} + C_{sch} N_{sch} + C_{uns} N_{uns}. \quad (13)$$

Here, N_{hv} , N_{sch} , and N_{uns} are the number of hangar visits, the number of scheduled replacements, unscheduled replacements, per year per aircraft, respectively. These values (N_{hv} , N_{sch} , and N_{uns}) are evaluated by Monte Carlo simulations (10^3 runs). Also, C_{set} , C_{sch} , and C_{uns} are the setup cost of a hangar visit, the cost of a scheduled replacement, and the cost of unscheduled replacement, respectively (see Section 3.1) for unscheduled replacements). The parameters C_{set} , C_{sch} and C_{uns} depend on the cost model of an aircraft operator. For this case study, we assume $C_{set} = 1$, $C_{sch} = 1$, and $C_{uns} = 2$.

The simulation results in Figure 8 and Table 3 show the benefit of utilizing RUL prognostics and considering component grouping, i.e., the benefit of the proposed RUL-driven OM. Figure 8 shows that the RUL-driven OM results in the lowest expected cost per aircraft per year. The results show that RUL-driven OM leads to 20% lower costs than PM, which is the traditional maintenance strategy.

Table 3 shows two reasons why the RUL-driven OM achieves the lowest expected cost. First, it has the smallest number of unscheduled replacements because it optimizes the mo-

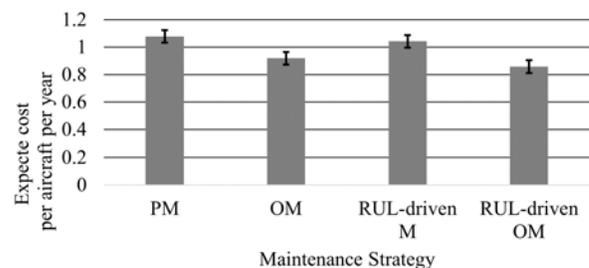


Figure 8. Expected cost and its 95% confidence interval.

Table 3. Performance of benchmarks.

	PM	OM	RUL-driven M	RUL-driven OM
Maintenance cost C	1.078	0.919	1.042	0.858
Scheduled replacements N_{sch}	0.255	0.327	0.291	0.370
Unscheduled replacements N_{uns}	0.223	0.154	0.186	0.111
Hangar visits N_{hv}	0.377	0.285	0.379	0.266

ment of replacements using RUL prognostics. Second, the RUL-driven OM results in the smallest number of hangar visits, saving the setup cost. Compared to the OM that minimizes the setup cost without considering RUL prognostics, the RUL-driven OM further reduces the number of hangar visits.

In Table 3, it is also interesting to see that the total number of scheduled and unscheduled replacements are roughly the same for all strategies, e.g., $N_{sch} + N_{uns} \approx 0.47$. This implies that the best maintenance strategy does not reduce the total number of replacements, but rather optimizes the timing of the replacements so that there is sufficient time to prepare tasks in advance, and reduce the setup cost.

5. CONCLUSION

In this study, we integrate Remaining-Useful-Life (RUL) prognostics for aircraft components into opportunistic maintenance planning that groups the maintenance of multiple components. First, the RULs of aircraft landing gear brakes are estimated based on a Bayesian regression model and the actual degradation data collected from a fleet of aircraft. Then, these prognostics are integrated into a maintenance planning optimization - opportunistic maintenance. With this, we group replacements of several brakes to reduce the setup cost for hangar visits. The proposed maintenance planning is applied for a long time horizon using a rolling horizon. Finally, the numerical results show that our proposed RUL-driven opportunistic maintenance planning results in a 20% reduction of total costs compared with several traditional maintenance strategies.

ACKNOWLEDGMENT

This research has partly been funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement No 769288.

REFERENCES

Aizpurua, J. I., Catterson, V. M., Papadopoulos, Y., Chiachio, F., & D’Urso, D. (2017). Supporting group maintenance through prognostics-enhanced dy-

namic dependability prediction. *Reliability Engineering and System Safety*, 168(April), 171–188. doi: 10.1016/j.ress.2017.04.005

Bouvard, K., Artus, S., Bérenguer, C., & Cocquempot, V. (2011). Condition-based dynamic maintenance operations planning & grouping. Application to commercial heavy vehicles. *Reliability Engineering and System Safety*, 96(6), 601–610. doi: 10.1016/j.ress.2010.11.009

de Jonge, B., & Scarf, P. A. (2020). A review on maintenance optimization. *European Journal of Operational Research*, 285(3), 805–824. doi: 10.1016/j.ejor.2019.09.047

de Pater, I., & Mitici, M. (2021). Predictive maintenance for multi-component systems of repairables with Remaining-Useful-Life prognostics and a limited stock of spare components. *Reliability Engineering and System Safety*, 214, 107761. doi: 10.1016/j.ress.2021.107761

de Pater, I., Reijns, A., & Mitici, M. (2022). Alarm-based predictive maintenance scheduling for aircraft engines with imperfect Remaining Useful Life prognostics. *Reliability Engineering & System Safety*, 221, 108341. doi: 10.1016/j.ress.2022.108341

Eleftheroglou, N., Mansouri, S. S., Loutas, T., Karvelis, P., Georgoulas, G., Nikolakopoulos, G., & Zarouchas, D. (2019). Intelligent data-driven prognostic methodologies for the real-time remaining useful life until the end-of-discharge estimation of the Lithium-Polymer batteries of unmanned aerial vehicles with uncertainty quantification. *Applied Energy*, 254(August), 113677. doi: 10.1016/j.apenergy.2019.113677

Kim, S., Choi, J.-h., & Kim, N. H. (2022). Inspection Schedule for Prognostics with Uncertainty Management. *Reliability Engineering and System Safety*, 108391. doi: 10.1016/j.ress.2022.108391

Lee, J., & Mitici, M. (2020). An integrated assessment of safety and efficiency of aircraft maintenance strategies using agent-based modelling and stochastic Petri nets. *Reliability Engineering and System Safety*, 202, 107052. doi: 10.1016/j.ress.2020.107052

Lee, J., & Mitici, M. (2022). Multi-objective design of aircraft maintenance using Gaussian process learning and adaptive sampling. *Reliability Engineering and System Safety*, 218, 108123. doi: 10.1016/j.ress.2021.108123

Mitici, M., & de Pater, I. (2021). Online model-based remaining-useful-life prognostics for aircraft cooling units using time-warping degradation clustering. *Aerospace*, 8(6). doi: 10.3390/aerospace8060168

Oikonomou, A., Eleftheroglou, N., Freeman, F., Loutas, T., & Zarouchas, D. (2022). Remaining Useful Life Prognosis of Aircraft Brakes. *International Journal of Prognostics and Health Management*, 13(1), 1–11. doi: 10.36001/ijphm.2022.v13i1.3072

- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... douard Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(1), 2825–2830. doi: 10.1145/2786984.2786995
- Pereira, D. P., Gomes, I. L., Melicio, R., & Mendes, V. M. (2021). Planning of aircraft fleet maintenance teams. *Aerospace*, 8(5). doi: 10.3390/aerospace8050140
- Saxena, A., & Goebel, K. (2008). *Turbofan Engine Degradation Simulation Data Set*. Moffett Field, CA: NASA Ames Research Center.
- Sprong, J. P., Jiang, X., & Polinder, H. (2019). A deployment of prognostics to optimize aircraft maintenance - A literature review. In *Proceedings of the annual conference of the prognostics and health management society, phm* (Vol. 11, pp. 1–12). doi: 10.36001/phm-conf.2019.v11i1.776
- van Noortwijk, J. M. (2009). A survey of the application of gamma processes in maintenance. *Reliability Engineering and System Safety*, 94, 2–21. doi: 10.1016/j.res.2007.03.019
- Vu, H. C., Do, P., Fouladirad, M., & Grall, A. (2020). Dynamic opportunistic maintenance planning for multi-component redundant systems with various types of opportunities. *Reliability Engineering and System Safety*, 198(July 2019), 106854. doi: 10.1016/j.res.2020.106854
- Wildeman, R. E., Dekker, R., & Smit, A. C. (1997). A dynamic policy for grouping maintenance activities. *European Journal of Operational Research*, 99(3), 530–551. doi: 10.1016/S0377-2217(97)00319-6
- Xia, T., Dong, Y., Pan, E., Zheng, M., Wang, H., & Xi, L. (2021). Fleet-level opportunistic maintenance for large-scale wind farms integrating real-time prognostic updating. *Renewable Energy*, 163, 1444–1454. doi: 10.1016/j.renene.2020.08.072

BIOGRAPHIES

Juseong Lee is a PhD candidate at the Air Transport and Operations Section, Faculty of Aerospace Engineering, Delft University of Technology. He holds an MSc degree in Aerospace Engineering from Korea Advanced Institute of Science and Technology (KAIST). His research interest includes the design and optimization of complex systems such as aircraft predictive maintenance.

Ingeborg de Pater is a PhD candidate at the Faculty of Aerospace Engineering, Delft University of Technology, the Netherlands. Her research interests are predictive aircraft maintenance scheduling and the Remaining Useful Life estimation of aircraft components.

Stan Boekweit has a MSc degree from Aerospace Engineering from the Air Transport and Operations Section, Faculty of Aerospace Engineering, Delft University of Technology.

Mihaela Mitici is an Assistant Professor at the Air Transport and Operations Section, Faculty of Aerospace Engineering, Delft University of Technology. She has a PhD in Stochastic Operations Research from University of Twente, the Netherlands. She specializes in Operations Research, with a focus on stochastic processes, decision-making under uncertainty, applied probability theory. Her main application domains are predictive aircraft maintenance, airport operations, and urban air mobility.

Novel Graph-Based Features for Bearing Fault Diagnosis: Two Aspects of Time-Series Structure

Sangho Lee¹, Chihyeon Choi², and Youngdoo Son³

^{1,2,3} *Dongguk University - Seoul, Seoul, Korea*
sangho218@dgu.ac.kr
choich0509@dgu.ac.kr
youngdoo@dongguk.edu

ABSTRACT

The feature-based methods for bearing fault diagnosis in prognostics and health management have been achieved satisfactory performances because of their robustness to noise and reduced dimension by pre-defined features. However, widely employed time- and frequency-domain features are insufficient to recognize the global pattern that indicates the structure of a time-series instance. In this paper, we propose two novel graph-based features which reflect the connection strength and degree of time series, respectively. First, we construct a graph of which an edge is defined as the Euclidean distance between each pair of time steps to measure the strengths of connections between the nodes. The other graph is constructed by the visibility algorithm, which converts a time series into a complex network to reflect the degrees of connections. Then, we calculate the Frobenius norms of the adjacency matrices of both graphs and use them as features for bearing fault diagnosis. To verify the proposed features, we performed several experiments with both synthetic and real datasets. From the synthetic datasets, it is observed that the changes in amplitudes and frequencies are detected by the features for the connection strength and degree, respectively. In addition, the proposed features also well-separate the distributions of each bearing state, including normal and several fault types, and show significant performance improvement as applied to the fault diagnosis task.

1. INTRODUCTION

As the complexity of equipment increases with industry development, the early detection of faults becomes important (Wei & Söffker, 2020). Feature-based methods for bearing fault diagnosis in prognostics and health management (PHM) have been achieved effective performances because of their robustness to noise and reduced dimension by pre-defined

features (Ma, Zheng, Li, & Cottrell, 2019). However, traditional features require a lot of domain knowledge and are specialized in time-domain and frequency-domain. Although the traditional features can reflect the local relationship of the time series to some extent, it is difficult to reflect the global pattern of the time series (Ferreira & Zhao, 2016). A graph is a powerful mechanism to recognize the global pattern of a time series by identifying the relationship between data points or groups (Ferreira & Zhao, 2016; Aminikhanghahi & Cook, 2017), so graph-based methods have been introduced to reflect structural information.

T. Li et al. (2020) converted frequency information of time-series signal into an affinity graph and performed the gearbox fault diagnosis by applying a modified graph convolutional network. Zhou et al. (2021) introduced a framework for constructing graphs from time-series signals and using it to rotating machinery fault diagnosis. C. Li et al. (2020) constructed a graph by applying a horizontal visibility algorithm (Luque, Lacasa, Ballesteros, & Luque, 2009) to a time series and performed bearing fault diagnosis using a graph neural network. Wang et al. (2019) constructed a graph by deriving a frequency spectrum based on periodogram estimation for normal data, and utilized it to detect bearing fault using statistical analysis. However, the previous studies require sufficient training data and domain knowledge to diagnose the faults accurately. If they are insufficient, it is difficult to diagnose the fault correctly.

Recently, to solve these limitations, some research based on spectral graph theory has emerged. The spectral graph theory is the study to recognize the properties of a graph, such as characteristic polynomial, eigenvalues, and eigenvectors of adjacency matrices associated with the graph. These studies aim to detect faults under the assumption that there are structure changes of data between the normal and fault states of machinery. The eigenvalues and eigenvectors are used to detect two aspects of structural changes in the graph. First, a community structure change, namely connection strength, oc-

Sangho Lee et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

curs when strongly connected nodes are weakened or weakly connected nodes become strongly connected. On the other hand, a change in community activity, namely connection degree, happens when the number of connections between nodes increases or decreases. We can identify the changes in connection strength and degree by monitoring eigenvalues and eigenvectors, respectively (Wang, Lu, Liu, & Yan, 2018; Kannan, Vempala, & Vetta, 2004; Sarkar & Boyer, 1998). With this approach, the methods based on spectral graph theory show good performance in terms of early fault detection and generalization with insufficient data owing to the cycle-to-cycle strategy.

Sun et al. (2020) used the spectral graph theory as pre-processing for feature extraction. Specifically, they introduced a method that extracts fault features using maximum correlated kurtosis deconvolution. To improve the performance of fault feature extraction, they firstly constructed an adjacency matrix by calculating Euclidean distance between time steps and used graph similarity based on eigendecomposition to identify fault states in advance. Lu et al. (2018) constructed adjacency matrices for time-series instances in a normal state in the same way as in Sun et al. (2020), and derived representative eigenvector and eigenvalue using eigendecomposition for the averaged matrix. Then, with the fixed derived eigenvector, they used a martingale-test based on the Frobenius norm of the difference of non-diagonal component between the derived eigenvalue matrix and that of a time-series instance to be tested. However, these methods only reflect the connection strength without consideration the presence or absence of connection degree between time steps and have high complexity, $\mathcal{O}(n^3)$, where n denotes the number of time steps in a time series, due to the use eigendecomposition of the adjacency matrix. Thus, it is difficult to use in practice. In addition, when calculating the eigenvector for the average of adjacency matrices, there is a risk of information loss on the raw time series.

Therefore, we propose simple graph-based features that relieve the limitations above. Euclidean distance and visibility algorithm (Lacasa, Luque, Ballesteros, Luque, & Nuno, 2008) are applied to time series to construct two adjacency matrices that reflect the connection strength and degree of a time-series instance, respectively, and the norms of these matrices are used as graph-based features. These features have the following advantages:

- It helps to achieve good performance in terms of early fault detection and generalization even with insufficient data owing to the cycle-to-cycle strategy of the spectral graph theory.
- Information loss that may occur from the average of adjacency matrices (Lu et al., 2018) is eliminated by analyzing the features of each time-series instance separately.
- Detection delay, which is inevitable for the cycle-to-

cycle strategy, is minimized by reducing the complexity of feature calculation.

- By reflecting not only the connection strength, but also the degree of connection between time steps, the structural information of the time series is sufficiently recognized.

2. PRELIMINARIES

In this section, we first explain traditional time- and frequency-domain features used to compare with the proposed features. Then, we briefly explain a visibility algorithm used to derive the connection degree of time series. Finally, Wasserstein and energy distances used to calculate the distance between class distributions are explained.

2.1. Time- and Frequency-Domain Features

Among the traditional time- and frequency-domain features presented in Jeon et al. (2015) and Jung et al. (2017), we used all (eight) time-domain features and chose three popular frequency-domain features that are the basis for calculating other frequency-domain features. The features used in this paper are presented in Table 1.

Table 1. The traditional time- and frequency-domain features used for fault diagnosis (Jeon et al., 2015; Jung et al., 2017). N is the number of time-series instances, t_n is a time-series instance, \bar{t} is the sample mean of all time-series instances, and σ denotes the standard deviation. f and $S(\cdot)$ denote the frequency and power spectrum function, respectively.

Domain	Feature	Description
Time	Kurtosis	$\sum_{n=1}^N \frac{(t_n - \bar{t})^4}{\sigma^4}$
	Skewness	$\sum_{n=1}^N \frac{(t_n - \bar{t})^3}{\sigma^3}$
	Absolute Mean	$\sum_{n=1}^N \frac{ t_n }{N}$
	Maximum	$\max(t_n)$
	RMS	$\sqrt{\sum_{n=1}^N \frac{ t_n ^2}{N}}$
	Crest factor	$\frac{\text{Maximum}}{\text{RMS}}$
	Shape factor	$\frac{\text{RMS}}{\text{Absolute Mean}}$
	Impulse factor	$\frac{\text{Maximum}}{\text{Absolute Mean}}$
Frequency	FC	$\frac{\int f \times S(f) df}{\int S(f) df}$
	RMSF	$\sqrt{\frac{\int f^2 \times S(f) df}{\int S(f) df}}$
	RVF	$\sqrt{\frac{\int (f - \text{FC})^2 \times S(f) df}{\int S(f) df}}$

The time-domain features are obtained by using the raw vibration signal itself. First, there are two data statistics-related

features, kurtosis, and skewness. The kurtosis is a feature of the sharpness of the data distribution and is used to measure how intensively the instances are centered. The skewness represents the degree of asymmetry of data distribution. In addition, there are three commonly included kinetic energy-related features, maximum, absolute mean, and root mean square (RMS). The maximum is the maximum value of the signal, the absolute mean is the average of the absolute magnitude of the signal, and the RMS denotes the most suitable feature to quantify the magnitude of the signal. Finally, three sinusoidal wave shape-related features calculated using the above time-domain features are also included in time-domain features. On the other hand, we used three fundamental frequency-domain features for comparison with the proposed features. Frequency center (FC) and root mean square frequency (RMSF) are the scales indicating the change in the position of the fundamental frequency, and root variance frequency (RVF) indicates the degree of cohesion of the power spectrum.

2.2. Visibility Algorithm

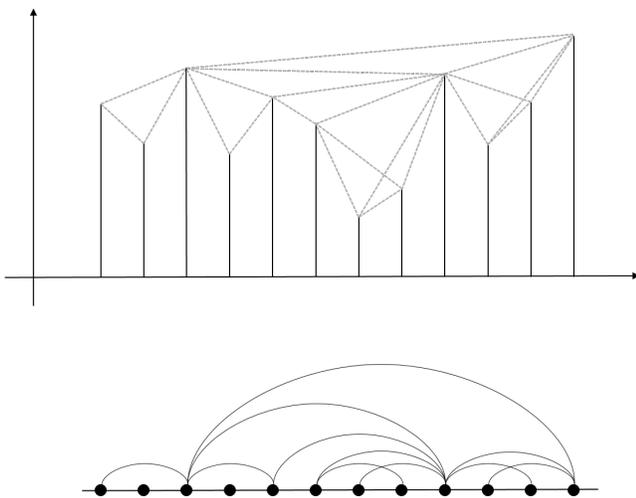


Figure 1. Overview of the visibility algorithm to obtain VG

Visibility algorithm converts a time series into a complex network (Lacasa et al., 2008). Figure 1 shows the overview of constructing visibility graph (VG) using the visibility algorithm. Several studies using VGs have proven that VGs help to effectively analyze time series and extract meaningful information (Y. Gao, Yu, & Wang, 2020; Z. Gao, Small, & Kurths, 2017). Let $T = \{t_1, t_2, \dots, t_n\}$ be the set of time steps in a time series, and $Y = \{y_1, y_2, \dots, y_n\}$ be the set of data values corresponding to time steps. Two arbitrary data points (t_a, y_a) and (t_b, y_b) are connected in a transformed graph, VG, if Eq. (1) is satisfied for all (t_c, y_c) placed between them.

$$y_c < y_b + (y_a - y_b) \times \frac{t_b - t_c}{t_b - t_a}, (a < c < b). \quad (1)$$

The resulting VG has the following properties (Lacasa et al., 2008):

- Connected: each adjacent node pair is connected
- Undirected: there is no directionality in the connection
- Invariant under affine transformations: the visibility criterion is invariant to affine transformations

In addition, we can recognize the time-series structure by analyzing the degree distribution of the resulting VG. For example, a periodic time series is converted into a regular graph, and a random time series are converted into an exponential random graph.

2.3. Distance Metrics for Distributions

In general, different classes (states) constitute separate manifolds. A feature that can well-distinguish each class distribution is a good feature that can provide useful information to a classifier (Bengio, Courville, & Vincent, 2013). To verify the usefulness of the proposed graph-based features, we measure the distances between class distributions derived by each feature. At this time, we use Wasserstein distance and energy distance to measure the distance between distributions (Arjovsky, Chintala, & Bottou, 2017; Shen, Qu, Zhang, & Yu, 2018; Bellemare et al., 2017).

The Wasserstein distance is a metric to measure the distance between two distributions and is defined as follows:

$$W(\mathbb{P}_X, \mathbb{P}_Y) = \inf_{\gamma \in \Pi(\mathbb{P}_X, \mathbb{P}_Y)} \mathbb{E}_{(x,y) \sim \gamma} \|x - y\|_1, \quad (2)$$

where $\Pi(\mathbb{P}_X, \mathbb{P}_Y)$ denotes the set of all possible joint distributions of which the marginal distributions are \mathbb{P}_X and \mathbb{P}_Y . The Wasserstein distance between \mathbb{P}_X and \mathbb{P}_Y is defined as the minimum expected value of $\|x - y\|_1$ (Arjovsky et al., 2017).

The other metric, energy distance, is also used to measure the distance between two distributions. When two objects are located in the same positions in the gravitational space, the potential energy between the two objects is zero, and the potential energy increases as the distance between the two objects increases. The energy distance extends this concept to measure the distance between two distributions. When X and Y are independent random vectors with cumulative distribution functions F and G , respectively, the square of the energy distance between F and G is defined as Eq. (3) (Rizzo & Székely, 2016).

$$D^2(F, G) = 2\mathbb{E}\|X - Y\| - \mathbb{E}\|X - X'\| - \mathbb{E}\|Y - Y'\|, \quad (3)$$

where $\|\cdot\|$ is Euclidean norm for each component, X' and Y' are random variables identical to X and Y , respectively, which are independent and identically distributed. The energy distance between F and G is calculated by the square root of $D^2(F, G)$.

3. PROPOSED FEATURES

In this paper, we propose two novel graph-based features, *variability* and *connectivity*, for two different aspects of time-series structure. The procedures of calculating two graph-based features are summarized in Algorithm 1.

Algorithm 1 Novel graph-based features

Input: A time-series instance $T = \{t_1, \dots, t_n\}$
Output: $\nu \in \mathbb{R}$: *variability*, $\kappa \in \mathbb{R}$: *connectivity*
 An Euclidean-based adjacency matrix $G \in \mathbb{R}^{n \times n}$
 A VG-based adjacency matrix $H \in \mathbb{R}^{n \times n}$
for $i, j = 1, \dots, n$ **do**
 Euclidean distance $d_{i,j} \leftarrow \|t_i - t_j\|$
 $G_{i,j} \leftarrow d_{i,j}$
 Assign visibility $h_{i,j}$ by visibility criterion (Eq. (1)).
 $H_{i,j} \leftarrow h_{i,j}$
end for
 $\nu \leftarrow \|G\|_F, \kappa \leftarrow \|H\|_F$

First, we obtain two adjacency matrices representing two different aspects, strength and degree of connection, of the time-series structural information. Specifically, given a time-series instance $T = \{t_1, \dots, t_n\}$, we derive an *Euclidean distance-based adjacency matrix* G corresponding to the time-series instance T , which is filled with the Euclidean distance values between all data point pairs ($\|t_i - t_j\|$). Thus, each element of G can represent the strength of the connection between data points. In addition, a VG-based adjacency matrix H corresponding to T is obtained. We apply the visibility criterion to each data point to construct H , so each element of H indicates whether t_i and t_j are connected.

Then, we introduce a formal definition of a graph-based feature, *variability*, that represents the strength of time-series connection.

Definition 1 (Variability) *Given an Euclidean distance-based adjacency matrix G corresponding to T , we define variability ν of T as*

$$\nu = \|G\|_F, \quad (4)$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

The *variability* feature, which is one aspect of the structural information of the time-series instance T , quantifies the overall strength that the data points in the time-series instance are connected. The Euclidean distance $d_{i,j}$, which is an element in the i -th row and j -th column of G , implies the connection strength between two data points t_i and t_j . Finally, the graph-based feature ν reflecting the overall connection strength of the time-series instance is extracted by calculating the Frobenius norm of G . The large *variability* value means that the Euclidean distances between all data points in the time-series instance are large; thus, we can conjecture that the time series has a large fluctuation.

On the other hand, we define a feature that reflects the degree of time-series connection as follows:

Definition 2 (Connectivity) *Given a VG-based adjacency matrix H corresponding to T , we define connectivity κ of T as*

$$\kappa = \|H\|_F, \quad (5)$$

where $\|\cdot\|_F$ is the Frobenius norm.

The *connectivity* feature that represents another aspect of the time-series structural information identifies how many connections that satisfy the visibility criterion exists between the data points in the time-series instance T . An element in the i -th row and j -th column of H is assigned 1 if data points t_i and t_j have visibility. Thus, it can represent the degree of connection for the time-series instance. Similar to deriving the *variability* feature ν , we reflect the overall connection degree of T by calculating the Frobenius norm for H . The variation of *connectivity* value means that degree of connection is changed, so it means that the structure of the time series is changed.

The time complexity of the calculation of *variability* is $\mathcal{O}(n^2)$ because the Euclidean distances between all data points in a time-series instance should be computed, and that of the *connectivity* is also $\mathcal{O}(n^2)$ due to complexity of visibility algorithm. Therefore, we can consider the global pattern of the time series, which is difficult to be recognized by the traditional time- and frequency-domain features, using the proposed graph-based features with scalable complexities.

4. EXPERIMENTS

We performed three experiments to analyze the properties of the two proposed features, *variability* and *connectivity*, with a synthetic dataset and to verify their usefulness and applicability for bearing fault diagnosis with the real-world dataset. In each experiment, the proposed features were compared with the traditional time- and frequency-domain features.

4.1. Data Description

First, we constructed three synthetic time series to analyze the properties of the proposed features. Since amplitude and frequency are important properties in the vibration signal (Wang et al., 2018), three synthetic time series were constructed in which amplitude and frequency changes exist. Each data point t_i of each synthetic time series T is composed as follows:

$$t_i = \begin{cases} \sin(2\pi i), & 1 \leq i < 2000, \\ 5 \times \sin(2\pi i), & 2000 \leq i \leq 3000. \end{cases}$$

$$t_i = \begin{cases} \sin(2\pi i), & 1 \leq i < 2000, \\ \sin(4\pi i), & 2000 \leq i \leq 3000. \end{cases}$$

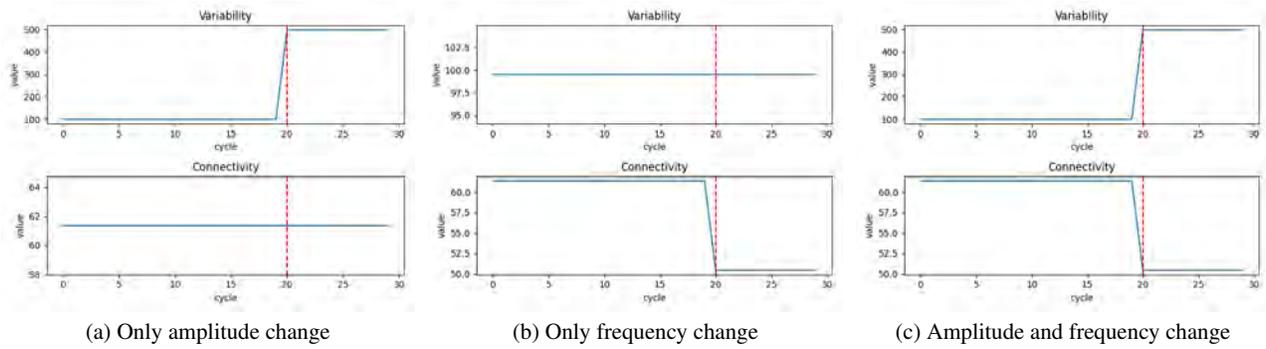


Figure 2. The values of each proposed feature corresponding to each cycle when (a) only amplitude change, (b) only frequency change, and (c) amplitude and frequency change. The x-axis represents the value of each feature, and the y-axis is the cycle.

$$t_i = \begin{cases} \sin(2\pi i), & 1 \leq i < 2000, \\ 5 \times \sin(4\pi i), & 2000 \leq i \leq 3000. \end{cases}$$

The first and second synthetic time series reflected the only amplitude and frequency changes, respectively. In the last synthetic time series, the amplitude and frequency changes were reflected simultaneously. For the experiment in Section 4.2, each synthetic time series was divided into 30 by setting the sequence length of a time-series instance to 100.

Next, to verify the usefulness and applicability of the proposed features for bearing fault diagnosis, we used the Case Western Reserve University (CWRU) bearing dataset (Smith & Randall, 2015), widely used as a benchmark dataset for various studies to identify bearing states (Chen, Mauricio, Li, & Gryllias, 2020; X. Li, Zhang, & Ding, 2019; Zhang et al., 2020; Afrasiabi, Afrasiabi, Parang, & Mohammadi, 2019). The CWRU bearing dataset contains vibration signals representing the operation states from bearings. The vibration signals were collected at 12 or 48 kHz for bearings under four types of motor loads (0, 1, 2, or 3 hp). The corresponding rotating speed for each motor load is 1797, 1772, 1750, or 1730 rpm. In addition, there are four bearing states: 1) normal, 2) inner race, 3) outer race, and 4) ball faults. Each fault state has various diameters (0.007, 0.014, or 0.021 inches). For the experiments in Sections 4.3 and 4.4, we only used the vibration signals collected from the drive end at 48 kHz and used faults with 0.007 fault diameter. Moreover, the corresponding sequence length for each motor load was calculated to approximate one rotation cycle dividing sampling frequency by the rotating speed; hence, we set each sequence length to 1610, 1630, 1650, or 1670.

4.2. Results of Property Analysis

We performed change detection using the synthetic dataset to analyze the properties of the proposed features. Figure 2 shows the changes of *variability* and *connectivity* values over-time for three synthetic time series.

The proposed features, *variability* and *connectivity*, can detect amplitude and frequency changes, respectively. For the synthetic time series with only amplitude change, the value of the *variability* feature changes at the time step where the amplitude changes, whereas the value of the *connectivity* is maintained. Conversely, in the synthetic time series with only frequency change, only the value of the *connectivity* feature changes according to the change in frequency. When both amplitude and frequency change, we can observe that both the values of *variability* and *connectivity* change at the time of change.

4.3. Results of Usefulness Verification

A feature that can provide useful information should have a well-distinguished distribution for each class (Bengio et al., 2013). Therefore, for the CWRU dataset, we calculated the distance between the class distributions formed by each feature. At this time, min-max scaling was applied to each feature to remove the influence of the scales of features. We used the Wasserstein and energy distances as the distance metrics. In this experiment, we used data with 0 hp of motor load. The results of the Wasserstein and energy distances are shown in Table 2 and Table 3, respectively.

The *variability* has a property that can detect amplitude change of the vibration signal. In general, there is a significant difference in amplitude between fault types; hence, we can observe that the *variability* had the largest distance between fault class distributions (BI, BO, and IO). In addition, the max and RMS, which belong to the time-domain features that can reflect amplitude information, had larger distances between fault class distributions than that of the frequency-domain features. Conversely, there is a considerable difference in frequency between normal and fault classes (NB, NI, and NO). Thus, the *connectivity*, which can detect the frequency change of vibration signal, had a third largest distance between normal and fault class distributions. In this case, most frequency-domain features also had larger dis-

Table 2. Wasserstein distance between the class distributions derived by each feature. The three largest average distances are highlighted in boldface. (N, normal state; B, ball fault; I, inner race fault; O, outer race fault; F, fault states; AVG., average)

Domain	Feature	NB	NI	NO	NF AVG.	BI	BO	IO	FF AVG.
Time	Kurtosis	0.61	0.05	0.76	0.47	0.56	0.15	0.71	0.47
	Skewness	0.13	0.07	0.28	0.16	0.08	0.16	0.23	0.16
	Abs. Mean	0.43	0.07	0.88	0.46	0.35	0.45	0.80	0.54
	Max	0.41	0.04	0.86	0.44	0.36	0.46	0.82	0.55
	RMS	0.38	0.06	0.90	0.45	0.32	0.51	0.84	0.56
	Crest factor	0.51	0.07	0.42	0.33	0.44	0.09	0.35	0.30
	Shape factor	0.00	0.00	0.24	0.08	0.00	0.24	0.24	0.16
	Impulse factor	0.01	0.00	0.27	0.09	0.01	0.26	0.27	0.18
Frequency	FC	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	RMSF	0.71	0.82	0.61	0.71	0.11	0.10	0.21	0.14
	RVF	0.46	0.70	0.90	0.69	0.24	0.44	0.20	0.29
Graph	Variability	0.38	0.06	0.90	0.44	0.32	0.52	0.84	0.56
	Connectivity	0.53	0.53	0.83	0.63	0.01	0.30	0.30	0.20

Table 3. Energy distance between the distributions of each state derived by each feature. The three largest average distances are highlighted in boldface. (N, normal state; B, ball fault; I, inner race fault; O, outer race fault; F, fault states; AVG., average)

Domain	Feature	NB	NI	NO	NF AVG.	BI	BO	IO	FF AVG.
Time	Kurtosis	1.05	0.18	1.17	0.80	0.99	0.35	1.12	0.82
	Skewness	0.22	0.14	0.55	0.30	0.15	0.35	0.47	0.32
	Abs. Mean	0.91	0.37	1.30	0.86	0.82	0.91	1.24	0.99
	Max	0.88	0.27	1.29	0.81	0.82	0.90	1.25	0.99
	RMS	0.87	0.33	1.32	0.84	0.79	0.97	1.27	1.01
	Crest factor	0.89	0.19	0.83	0.64	0.82	0.22	0.75	0.60
	Shape factor	0.05	0.05	0.59	0.23	0.08	0.59	0.60	0.42
	Impulse factor	0.11	0.03	0.63	0.25	0.12	0.62	0.63	0.46
Frequency	FC	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	RMSF	1.14	1.23	1.06	1.14	0.33	0.38	0.60	0.44
	RVF	0.92	1.15	1.32	1.13	0.65	0.92	0.60	0.72
Graph	Variability	0.86	0.32	1.32	0.83	0.79	0.98	1.27	1.01
	Connectivity	0.53	0.53	0.83	1.08	0.01	0.30	0.30	0.48

Table 4. Accuracy (%) of the model trained various feature combinations of the time-, frequency-domain, and graph-based features. In each motor load, the best performance is highlighted in boldface.

Motor Load (hp)	Feature Combination						
	Time	Frequency	Graph	Time & Frequency	Time & Graph	Frequency & Graph	All
0	96.11	21.67	99.72	96.67	99.72	99.72	99.44
1	97.78	21.67	100.00	98.06	100.00	100.00	100.00
2	97.22	25.00	97.22	93.89	97.22	97.22	97.22
3	97.50	25.83	100.00	97.78	100.00	100.00	100.00

tances than the time-domain features. In summary, similar to the time- and frequency-domain features, the *variability* and *connectivity* can represent amplitude and frequency information of the vibration signal using two aspects of time-series structure, respectively. Moreover, we confirmed that the proposed features well-distinguished class distributions accord-

ing to their properties demonstrated in Section 4.2.

4.4. Results of Applicability Verification

To verify that the proposed graph-based features, *variability* and *connectivity*, are adequate for bearing fault diagnosis, we

performed a classification task and compared the results of the model trained with various feature combinations, including the time-, frequency-domain, and graph-based proposed features. We used logistic regression as a classifier because it does not require additional parameter tuning. Table 4 shows the accuracy of each model, which was derived with a one-vs-rest strategy. In this experiment, we constructed a dataset with 30 instances randomly sampled per class, and 70% of the dataset was used to train the model and the rest to test model performance. To reduce the effect of randomness, we repeated the procedure ten times and reported the averaged results across all runs.

Although the number of the graph-based features is smaller than that of the other domains, the trained model only with the proposed features showed better performance than the trained model only with the traditional features, regardless of motor loads. Furthermore, when the traditional and proposed features were used together to train the model, the performance was improved compared to training the model only with the traditional features. It can be explained that the proposed features, *variability* and *connectivity*, play an important role in bearing fault diagnosis by reflecting structural information of time series, amplitude and frequency, while the traditional features only provide redundant information.

5. CONCLUSION

We propose novel graph-based features, *variability* and *connectivity*, for reflecting structural information of time series. We construct two graphs using the Euclidean distance and visibility algorithm and obtain the proposed features by calculating Frobenius norms of their adjacency matrices. Through several experiments on synthetic and real bearing datasets, we demonstrated that the *variability* and *connectivity* could reflect amplitude and frequency information, respectively, with reasonable complexities and well-separate the distributions of bearing states. The model trained only with the proposed features achieved significant performance in the bearing fault diagnosis task. Moreover, the proposed features helped improve the model performance trained with the other domain features. Therefore, the *variability* and *connectivity* features are useful features to classify bearing states in fault diagnosis.

ACKNOWLEDGMENT

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT: Ministry of Science and ICT) (Nos. 2020R1C1C1003425 and 2020R1A4A3079710).

REFERENCES

Afrasiabi, S., Afrasiabi, M., Parang, B., & Mohammadi, M.

- (2019). Real-time bearing fault diagnosis of induction motors with accelerated deep learning approach. In *2019 10th international power electronics, drive systems and technologies conference (pedstc)* (pp. 155–159).
- Aminikhanghahi, S., & Cook, D. J. (2017). A survey of methods for time series change point detection. *Knowledge and information systems, 51*(2), 339–367.
- Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning* (pp. 214–223).
- Bellemare, M. G., Danihelka, I., Dabney, W., Mohamed, S., Lakshminarayanan, B., Hoyer, S., & Munos, R. (2017). The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*.
- Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence, 35*(8), 1798–1828.
- Chen, Z., Mauricio, A., Li, W., & Gryllias, K. (2020). A deep learning method for bearing fault diagnosis based on cyclic spectral coherence and convolutional neural networks. *Mechanical Systems and Signal Processing, 140*, 106683.
- Ferreira, L. N., & Zhao, L. (2016). Time series clustering via community detection in networks. *Information Sciences, 326*, 227–242.
- Gao, Y., Yu, D., & Wang, H. (2020). Fault diagnosis of rolling bearings using weighted horizontal visibility graph and graph fourier transform. *Measurement, 149*, 107036.
- Gao, Z., Small, M., & Kurths, J. (2017). Complex network analysis of time series. *EPL (Europhysics Letters), 116*(5), 50001.
- Jeon, B. C., Jung, J. H., Youn, B. D., Kim, Y., & Bae, Y. (2015). Datum unit optimization for robustness of a journal bearing diagnosis system. *International Journal of Precision Engineering and Manufacturing, 16*(11), 2411–2425.
- Jung, J. H., Jeon, B. C., Youn, B. D., Kim, M., Kim, D., & Kim, Y. (2017). Omnidirectional regeneration (odr) of proximity sensor signals for robust diagnosis of journal bearing systems. *Mechanical Systems and Signal Processing, 90*, 189–207.
- Kannan, R., Vempala, S., & Vetta, A. (2004). On clusterings: Good, bad and spectral. *Journal of the ACM (JACM), 51*(3), 497–515.
- Lacasa, L., Luque, B., Ballesteros, F., Luque, J., & Nuno, J. C. (2008). From time series to complex networks: The visibility graph. *Proceedings of the National Academy of Sciences, 105*(13), 4972–4975.
- Li, C., Mo, L., & Yan, R. (2020). Rolling bearing fault diagnosis based on horizontal visibility graph and graph neural networks. In *2020 international conference on sensing, measurement & data analytics in the era of*

artificial intelligence (icsmd) (pp. 275–279).

- Li, T., Zhao, Z., Sun, C., Yan, R., & Chen, X. (2020). Multireceptive field graph convolutional networks for machine fault diagnosis. *IEEE Transactions on Industrial Electronics*, 68(12), 12739–12749.
- Li, X., Zhang, W., & Ding, Q. (2019). Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism. *Signal processing*, 161, 136–154.
- Lu, G., Liu, J., & Yan, P. (2018). Graph-based structural change detection for rotating machinery monitoring. *Mechanical Systems and Signal Processing*, 99, 73–82.
- Luque, B., Lacasa, L., Ballesteros, F., & Luque, J. (2009). Horizontal visibility graphs: Exact results for random time series. *Physical Review E*, 80(4), 046103.
- Ma, Q., Zheng, J., Li, S., & Cottrell, G. W. (2019). Learning representations for time series clustering. *Advances in neural information processing systems*, 32, 3781–3791.
- Rizzo, M. L., & Székely, G. (2016). Energy distance. *wiley interdisciplinary reviews: Computational statistics*, 8(1), 27–38.
- Sarkar, S., & Boyer, K. L. (1998). Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. *Computer vision and image understanding*, 71(1), 110–136.
- Shen, J., Qu, Y., Zhang, W., & Yu, Y. (2018). Wasserstein distance guided representation learning for domain adaptation. In *Thirty-second aai conference on artificial intelligence*.
- Smith, W. A., & Randall, R. B. (2015). Rolling element bearing diagnostics using the case western reserve university data: A benchmark study. *Mechanical systems and signal processing*, 64, 100–131.
- Sun, W., Zhou, Y., Cao, X., Chen, B., Feng, W., & Chen, L. (2020). A two-stage method for bearing fault detection using graph similarity evaluation. *Measurement*, 165, 108138.
- Wang, T., Lu, G., Liu, J., & Yan, P. (2018). Graph-based change detection for condition monitoring of rotating machines: Techniques for graph similarity. *IEEE Transactions on Reliability*, 68(3), 1034–1049.
- Wang, T., Lu, G., & Yan, P. (2019). A novel statistical time-frequency analysis for rotating machine condition monitoring. *IEEE Transactions on Industrial Electronics*, 67(1), 531–541.
- Wei, X., & Söffker, D. (2020). Comparison of cwru dataset-based diagnosis approaches: Review of best approaches and results. In *European workshop on structural health monitoring* (pp. 525–532).
- Zhang, J., Yi, S., Liang, G., Hongli, G., Xin, H., & Hongliang, S. (2020). A new bearing fault diagnosis method based on modified convolutional neural networks. *Chinese Journal of Aeronautics*, 33(2), 439–447.
- Zhou, K., Yang, C., Liu, J., & Xu, Q. (2021). Dynamic graph-based feature learning with few edges considering noisy samples for rotating machinery fault diagnosis. *IEEE Transactions on Industrial Electronics*.

BIOGRAPHIES



Sangho Lee was born in Seoul, Republic of Korea in 1995. He received B.S. and M.S. degrees in Industrial and Systems Engineering from Dongguk University - Seoul, Seoul, Korea, in 2018 and 2020, respectively. He is currently pursuing the Ph.D. degree in Industrial and Systems Engineering at the Dongguk University-Seoul. His research interests include time-series analysis, complex network theory, machine learning, and their applications to PHM.



Chihyeon Choi was born in Busan, Korea in 1996. He received B.S. degrees in Industrial and Systems Engineering from Dongguk University - Seoul, Seoul, Republic of Korea, in 2022. He is currently pursuing the M.S. degree in Industrial and Systems Engineering at the Dongguk University-Seoul. His research interests include time-series analysis, learning with noisy labeled data, machine learning, and their applications to industrial problems.



Youngdoon Son is an assistant professor at the department of Industrial and Systems Engineering, Dongguk University - Seoul, Seoul, Korea. He received B.S. in Physics and M.S. in Industrial and Management Engineering from Pohang University of Science and Technology (POSTECH), Gyeongbuk, Republic of Korea in 2010 and 2012, respectively, and Ph.D. in Industrial Engineering from Seoul National University in 2015. His research interests include machine learning and data analytics, and their applications to industrial problems.

Certainty Groups: A practical approach to distinguish confidence levels in neural networks

Lukas Lodes¹, Alexander Schiendorfer²

^{1,2} *Technische Hochschule Ingolstadt, Germany*

Lukas.Lodes@thi.de

Alexander.Schiendorfer@thi.de

ABSTRACT

Machine Learning (ML), in particular classification with deep neural nets, can be applied to a variety of industrial tasks. It can augment established methods for controlling manufacturing processes such as statistical process control (SPC) to detect non-obvious patterns in high-dimensional input data. However, due to the widespread issue of model miscalibration in neural networks, there is a need for estimating the predictive uncertainty of these models. Many established approaches for uncertainty estimation output scores that are difficult to put into actionable insight. We therefore introduce the concept of certainty groups which distinguish the predictions of a neural network into the normal group and the certainty group. The certainty group contains only predictions with a very high accuracy that can be set up to 100%. We present an approach to compute these certainty groups and demonstrate our approach on two datasets from a PHM setting.

1. INTRODUCTION

Modern production is a complex interaction of different parts and optimal usage of available production resources is hard to accomplish. Data analytics is used for decades to support decision making and gained additional attention in the last years through machine learning with deep neural networks (DNN). The potential of DNN-based classification is increasingly being explored in the context of prognostics and health management (PHM). There exists a wide variety of promising applications, including the ideal timing of maintenance intervals (predictive maintenance) or the prediction of health indices of the production line itself or of the products. Despite the availability of the technology, these tools are still not commonly used in the manufacturing industry. Besides the slow adoption of digitization in the manufacturing indus-

try in general, there are further reservations about machine learning systems in critical applications like quality inspection. One problem are difficulties in understanding and judging the model's outputs, in particular, if they represent a probability distribution over a finite number of classes, as returned by a softmax or sigmoid activation. For neural networks, the widespread issue of model miscalibration (Guo, Pleiss, Sun, & Weinberger, 2017) leads to predictions in which the model assigns a high probability score to a wrong class. This makes it hard to trust the predictions of such a model. Even if a predictive ML model errs on some cases, it should provide a reliable estimate of its uncertainty in order to mitigate potential negative impact of false predictions. In a production system, this property could be used for *Hybrid Quality Inspection* (Ismail, Mostafa, & El-assal, 2021), in which automated quality inspection of the products is done only for a subset of all product instances – those considered difficult to judge would still undergo manual, i.e., human-operated tests. As stated, the ability of a neural network to express its predictive uncertainty is crucial for real world application. In the last few years, increasing attention has been paid to the field of uncertainty estimation in neural networks (an overview is given in Section 1.1). There exist several different approaches to uncertainty estimation in neural networks, and many of them share a common principle. The predictive uncertainty is often expressed by several metrics (usually mean and standard deviation) that are calculated on a sample of predictions done by slightly different estimators. These metrics however are hard to put into actionable insight as they are still relatively abstract. From the point of view of user-friendliness, a clear and concise classification of whether a prediction is based on a high or low uncertainty would be highly desirable. This is especially important in situations in which the estimated uncertainty of a prediction has to be converted into a binary decision, e.g., by a human worker in a production line. Additionally, many methods for estimating the predictive uncertainty in neural networks tend to involve a very large theoretical background (e.g., fully Bayesian methods involving a meticulous choice of priors) that further complicates the ap-

Lukas Lodes et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

plication of these methods in applied machine learning applications such as predicting the health status of equipment or the quality of products. Thus, ease of use and manageability were further criteria in the development of our approach.

We therefore propose the concept of *certainty groups* which divides the predictions of a neural network into the normal group and the certainty group. The certainty group contains only predictions (and, consequently, instances) with a very high accuracy that can be set up to 100%. We assume that these instances are handled by an autonomous ML-system sufficiently well and don't require further action (e.g., established quality assurance methods). However, the other cases are potentially unsafe to be fully processed by an ML system and need to be more thoroughly examined. These instances represent the normal group, in which the predictions behave like usual, with them being correct with a certain accuracy. This leads to a division of the data set in different levels of model confidence. In the underlying paper, we present an approach based on ensembles to compute these certainty groups by measuring the disagreement between the ensemble members and sorting it based on a threshold value. We validate our approach using two datasets from a PHM setting, in which the method generates groups of samples that contain 26-51% of all instances with very few mispredictions compared to baseline models and approaches. We further show that our approach is highly flexible as it allows to configure the quality of the predictions contained in the certainty group.

1.1. Related Work

Due to the renewed interest in deep learning after breakthroughs in image classification or reinforcement learning, neural networks have received a lot of attention in the last 5 to 10 years. Therefore it's not surprising that there exist many recent studies that cover the application of neural networks in PHM related tasks like remaining useful life (RUL) prediction or predictive maintenance in general. Comprehensive surveys and reviews are provided in the works of (Zhao et al., 2016) and (Zhang et al., 2019). (Harshavardhanan & Nene, 2020) outlines why the consideration of uncertainty is beneficial when making predictions in a PHM related setting by explaining common sources of uncertainty and giving an overview on how to incorporate uncertainty into PHM systems. The area of uncertainty estimation in neural networks has been actively researched in the last couple of years. The first branch of works (see, e.g., (Guo et al., 2017; Tomani & Buettner, 2021; Naeini, Cooper, & Hauskrecht, 2015)) tries to directly solve the issue of miscalibration, either by post-processing a trained model or already during training (e.g. through additional loss functions). The second branch is based on the idea of ensembles, which are an established approach for estimating uncertainty in general and for obtaining more robust predictions (Lee, Purushwalkam, Cogswell, Crandall, & Batra, 2015; Wen, Tran, & Ba, 2020a; Havasi et al., 2020).

These works aim to further develop classical ensembling by increasing the accuracy of the ensemble prediction or improving the computational performance, which they achieve by different types and degrees of parameter sharing between the ensemble members. The works of (Daxberger et al., 2021) and (Wen, Tran, & Ba, 2020b) are examples for the integration of Bayesian methods in neural networks. (Daxberger et al., 2021) suggests an easy to use approach for converting conventional neural networks to bayesian neural networks by adding laplace approximation to the final layer, while (Wen et al., 2020b) achieves distance aware outputs by adding a gaussian process to the network. Additionally, in the last years a variety of frameworks for uncertainty estimation were published. These frameworks cover a wide scope, ranging from implementations of neural network based approaches like Deep Ensembles (e.g. in (Weiss & Tonella, 2021)) to fully fledged probabilistic programming languages (Tehrani et al., 2020; Salvatier, Wiecki, & Fonnesbeck, 2016; Bingham et al., 2018; Phan, Pradhan, & Jankowiak, 2019) The mentioned works provide approaches for uncertainty estimation well-founded in probability theory – which also requires the selection of proper priors and sampling or variational inference for approximation. Our approach draws upon these methods and adds an additional layer of abstraction in order to improve the usability of uncertainty estimates.

2. APPROACH

As mentioned in the introduction, our approach is mainly motivated by creating an easy-to-use method with understandable outputs for practitioners in manufacturing. The targeted application area of our approach are production lines in which workers will have to work with the recommendations of a ML system. In the scope of this paper, we focus on binary classification problems – “OK” vs. “NOK” for a product, or “Machine defect” vs. “Machine intact” for health state estimation. We feel that for this use case the output of a ML system should be as easy as possible to understand to create transparent decision criteria and to make the workers more comfortable working with such a system. That's why we decided to provide again a binary decision whether a prediction is afflicted with uncertainty or whether the system is very certain about it. For a given instance x , we then have to first decide whether it falls into the certainty group or not and, second, if it belongs to class 0 or 1. The latter prediction is trusted more if x belongs to the certainty group.

2.1. Foundations

In the following, we consider a neural network model m having the form $f_m(x; \theta)$ where θ refers to the parameters that are optimized with respect to data set $D = (x_i, y_i)_{i=1}^N$ to obtain a point estimate (i.e., empirical risk minimization or maximum likelihood estimation). The output is assumed to be normalized, i.e., it is a real number that represents the

probability that x belongs to class 1:

$$p(y = 1 | x) = f_m(x; \theta) \quad (1)$$

In the case of neural networks, this is the direct output e.g. of a sigmoid or softmax layer. For example, $f_m(x; \theta) = 0.8$ represents estimating a probability of 80 % for class 1 and 20 % for class 0. In accordance with the literature (Guo et al., 2017), we refer to these normalized raw score outputs as *confidences*. The rounded confidences, i.e., $y = 0$ if $f_m(x; \theta) \leq 0.5$ and $y = 1$ otherwise, are called the *predictions*. A mismatch between a model's output confidence and the true correctness likelihood is called *miscalibration* (Guo et al., 2017). For example, miscalibration occurs if a model only reaches 60% accuracy on all test samples that it predicted to be 90% sure. The evaluation of a neural network m , i.e., calculating $f_m(x; \theta)$ for a particular input x and fixed parameters θ , is called *inference* (or, often, simply a forward pass) to distinguish the training from the productive stage.

In addition, we assume a strategy for uncertainty estimation $\zeta_m \geq 0$ where $\zeta_m(x | f_m, \theta) = 0$ means absolute certainty that $p(y = 1 | x)$ is indeed $f_m(x; \theta)$ and larger values indicate higher uncertainty. For example, given an ensemble of models f_{m_i} that are derived from m , ζ_m could refer to the sample standard deviation of the confidences $f_{m_i}(x; \theta)$ or other dispersion metrics. Other measures ζ_m could be retrieved from a Bayesian approach (i.e., having estimated a posterior over the parameter space $p(\theta | D)$) as the variance of the posterior predictive distribution

$$p(y = 1 | D, x) = \int_{\Theta} p(y = 1 | \theta, D, x) p(\theta | D, x) d\theta \quad (2)$$

which might not be tractable but only approximated via sampling or variational methods. In principle, our approach targets both point estimates with sample statistics as well as Bayesian techniques – although the high computational costs of Bayesian inference leads us to turn to practically applicable sample-based methods.

2.2. Certainty Groups: A general definition

Our goal is to define certainty groups that isolate the instances for which the model's predictions are very reliable from the rest. In a manufacturing setting, these would be the parts that are definitely detected as scrap or good. Informally speaking, the *certainty group* of a model is created by an uncertainty estimator and a predicate that decides whether the estimated uncertainty of an instance's prediction is low enough. The instances whose estimated uncertainty is too high to be in the Certainty Group are contained in the *normal group*. They are usually significantly harder to classify and, thus, the model's predictions become less accurate. Returning to the above example, these would be the parts that have to undergo further testing and human intervention – as the model is not capable

of determining their state. This leads to a binary distinction of the confidence levels in the underlying neural network. Figure 1 illustrates this concept.

In order to allow for a wide variety of uncertainty metrics ζ_m and to allow future extensions, we decided to define the Certainty Group of a model in a very open manner.

The certainty group CG of a model m with respect to data set D is then defined as follows:

$$CG(m, D) = \{(x, y) \in D \mid \varphi \text{ accepts } \zeta_m(x | f_m, \theta)\} \quad (3)$$

where φ is a boolean predicate (called the certainty criterion) that can consist of the operators \wedge (AND), \vee (OR), \neg (NOT), $<$, \leq , $>$ and \geq . The specific predicate depends on the chosen approach to estimate the model's uncertainty and is further described later in this section.

2.3. Possible uncertainty measures for certainty groups

We tested two principles for computing certainty groups from model confidences and predictions. The first, main, approach uses ζ_m along with some thresholding for some model m to partition the instances into normal and certainty group. Along the way, some methods require to (temporarily) generate a finite number of models m_i derived from m (e.g., via different dropout masks) to calculate ζ_m based on the confidences and predictions retained from the individual m_i . We call these methods *sample-based*. The second, baseline, approach simply sets $\zeta_m = f_m$, i.e., it uses the raw confidences as an indicator for certainty – e.g., interpreting a model output $f_m(x; \theta) = 0.99$ to truly be 99% sure.

An uncertainty estimator can obtain the prediction y and standard deviation σ for an instance x as follows (note that this requires a model that implements a strategy for estimating uncertainty through multiple model outputs): First, it obtains n samples of the model's prediction for the instance, that can be formally described as $p_\theta(\hat{y} | x)$.

A sample-based uncertainty estimator first generates n samples, i.e., model confidences $(f_{m_i}(x; \theta))_{i \in \{1, \dots, n\}}$ for a given instance x , e.g., $[0.85, 0.9, 0.92, \dots]$.

Then it calculates the sample mean \bar{f} and sample standard deviation s on these n sampled confidences and calculates the final class prediction y based on the sampled confidences by rounding \bar{f} . Finally, the decision whether this instance and prediction is contained in the certainty group is made by comparing s to an empirically determined threshold hyperparameter T_s . Referring to equation 3, this amounts to a straightforward predicate φ . In this case, the prediction is part of CG if

$$s \leq T_s \quad (4)$$

It is possible to calculate \bar{f} and s based on the individual models' predictions (rounded confidence values) as well, but the

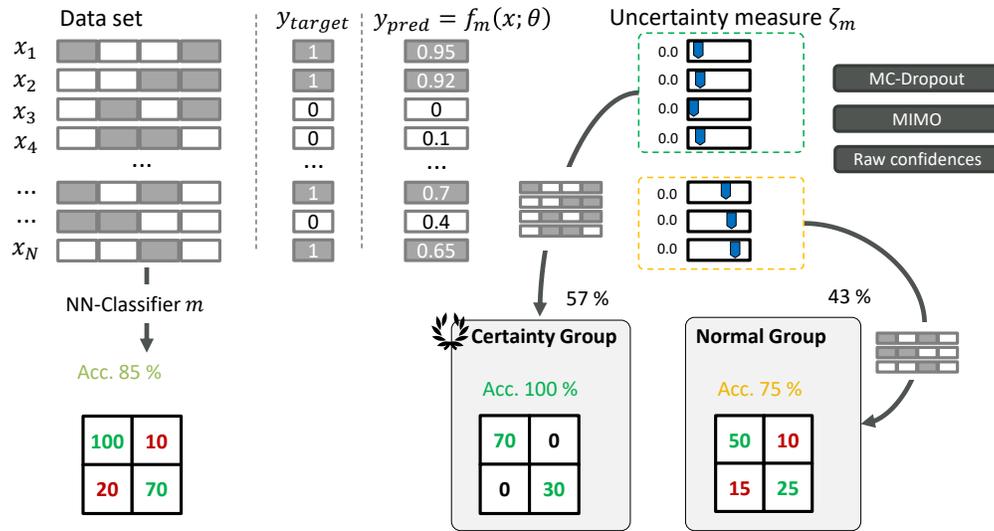


Figure 1. The predictions of a neural network m are evaluated in their uncertainty by an uncertainty measure ζ_m . If a prediction has a low enough uncertainty, it is assigned to the so called Certainty Group. The less-certain predictions of m are contained in the Normal Group. This usually leads to a significantly higher accuracy in the Certainty Group than in the Normal Group.

usage of raw confidence values allows for a much finer resolution w.r.t. φ , which we found to lead to larger certainty group sizes in our experiments, i.e., it was a more distinctive certainty criterion.

The introduction of the additional hyperparameter T_s allows for flexibility in terms of predictive accuracy in the certainty group. While we were initially aiming towards an accuracy of 100% in the certainty group, it's easily possible to lower the requirements for accuracy if the application allows it to get larger certainty group sizes.

The open definition of certainty groups in equation 3 allows us to define the baseline that only relies on the raw confidences. The most naive approach for computing certainty groups is to accept the confidence margins around 0 and 1 as safe instances and therefore include instances with model confidences in these margins in the certainty group. Formally speaking, an instance x is in CG for model m if

$$f_m(x; \theta) \leq T_{lower} \vee f_m(x; \theta) \geq T_{upper} \quad (5)$$

where T_{lower} and T_{upper} are two thresholds that define the border of the lower and upper decision margin.

2.4. Specific approaches to obtain ensembles from models

As stated in section 1.1, there is a wide variety of approaches that can be used either on top of existing neural networks or directly included in the architecture. From these approaches we chose a Dropout approach according to (Gal & Ghahramani, 2016) and multi-input multi-output configurations (MIMO) by (Havasi et al., 2020) as they fit our requirement for ease of use and manageability very well. In the following, we

describe both approaches and discuss them to further justify our decision for these two approaches.

Dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) is a widely used regularizer in neural networks and is used to make a neural network more robust against overfitting the training data set. The idea of this approach is to randomly drop connections in a neural network with a certain probability, which leads to a slightly different network configuration in each inference step (i.e., forward pass or evaluation of the network). Gal et al. describe this mathematically by sampling binary variables (i.e., from a Bernoulli distribution) for every unit in the neural network with probability p_i for value 1 in each layer L_i except for the output layer. If the corresponding binary variable takes value 0, that particular unit is dropped by setting it to zero. Usually, dropout is only activated during training and deactivated in the test or deployment stage (once training finished). Gal et al. however suggested the activation of dropout during the test or deployment stages to enable a Bayesian approximation without the need for computationally expensive implementations of Bayesian neural networks. With activated dropout during model inference, the model's uncertainty can be estimated by doing n distinct inference passes – each corresponding to a different network configuration of deactivated connections. That way, we can obtain a sample-based uncertainty estimator ζ_m . Mathematically speaking, the goal is to calculate the predictive distribution $p(y | x, D)$ for a new input point x^* given the data set D . As this predictive distribution cannot be obtained analytically, it is approximated by the approximate predictive distribution $q(y | x, D)$. The authors show that the expected value of this distribution (serving as the model's confidence)

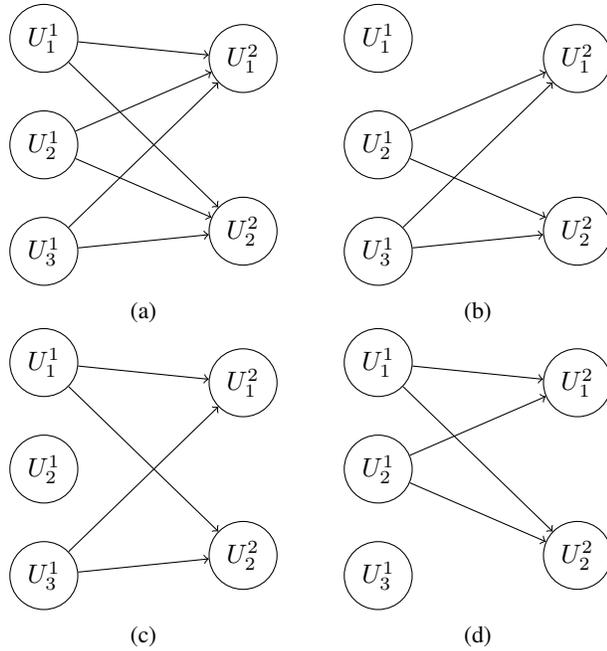
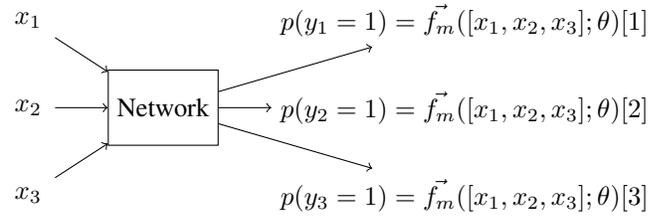


Figure 2. Visualization of MC Dropout with $T = 3$ inference passes in a network with 2 layers where U_i^n describes the i -th unit in layer n . (b), (c) and (d) show three different realizations of the base architecture shown in (a).

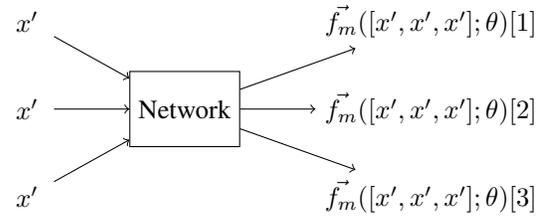
can be estimated by

$$\mathbb{E}_q(f_m) \approx \frac{1}{n} \sum_{i=1}^n f_{m_i}(x; W_1^i, \dots, W_L^i) \quad (6)$$

where m is a neural network with L layers and $\{W_1^i, \dots, W_L^i\}_{i=1}^n$ are the random variables representing the trainable parameters θ of a neural network with dropout. These random variables contain the actual dropout mask on the neural network's parameters. The authors refer to this Monte Carlo estimate as *MC Dropout*. As the activation of dropout during inference results in slightly different model architectures at each inference pass, it can be interpreted as an ensemble with n members $(f_{m_i})_{i=1}^n$ when calculating n inference passes. This process is illustrated in Figure 2. We chose this approach because dropout is already widely used and it is convenient to obtain an uncertainty estimate from this approach. Frameworks like PyTorch or Tensorflow allow an activation of dropout during the inference phase, which greatly reduces the required implementation effort. The most significant downside of this approach is the slow inference speed. As we need to compute n inference passes to create an ensemble with size n (and generally require large enough n to acquire reliable estimates), the approach is approximately n times slower than approaches that require only one inference pass. Our approach builds upon this method by implementing an actual decision rule for when to accept a prediction as ‘‘certain’’ and when as ‘‘uncertain’’. Gal et al. roughly propose the need for such a



(a) In the train phase the network input consists of a concatenation of n instances. The network outputs n concatenated confidences corresponding to the different input instances.



(b) In the test phase the network input consists of n copies of an unseen input instance x . The network outputs n different confidences for the input instance.

Figure 3. Illustration of the MIMO configuration proposed by Havasi et al.

decision rule in the context of classification.

Similar to MC dropout, the usage of MIMO configurations from Havasi et al. aims towards the implicit creation of an ensemble. The foundation of this approach is given by the results of works covering sparsity in modern neural networks (Frankle & Carbin, 2018; Molchanov, Tyree, Karras, Aila, & Kautz, 2016; Zhu & Gupta, 2017). These works show that modern neural networks often are overparametrized for their tasks, i.e. they often could solve the task multiple times in terms of capacity. Havasi et al. use this assumption to concurrently train multiple independent subnetworks within one single larger network using the proposed multi-input, multi-output (MIMO) configuration. For this approach to work, only two small changes have to be made to an existing model architecture. In contrast to the normally used model inputs that consist of a single instance, n input instances are concatenated into one single instance. In this case, n represents the number of ensemble members contained inside the network. This can be formally described in the following way. Given a data set $D = (x_i, y_i)_{i=1}^N$ with size N where x_i is an instance with corresponding label y_i . Usually, the model output is given by $f_m(x_i; \theta)$. In the MIMO configuration however, the model input is given by a concatenation of n inputs $\{x_1, \dots, x_n\}$. The model output does not consist of only one so called *head*, but of n *heads* that output n confidences as a

vector \vec{f}_m :

$$\vec{f}_m = (f_{m_1}([x_1, \dots, x_n]; \theta), \dots, f_{m_n}([x_1, \dots, x_n]; \theta)) \quad (7)$$

In the training phase, the concatenated input instance consists of different instances, which leads to the creation of independent subnetworks that are contained inside the large network. During test time or inference in the deployment stage, the concatenated input instance is created by duplicating one unseen test input instance x' n times, so $x_1 = \dots = x_n = x'$. This leads to n output confidences for the same instance. This way, one can obtain an ensemble with n members in only one inference pass, assuming that the overall model has enough capacity (i.e., parameters θ) to solve the given task n times. With respect to our sample-based uncertainty metrics, we consider the components of the output vector as individual derived confidences, i.e., $\vec{f}_m(x; \theta)[i] \hat{=} f_{m_i}(x; \theta)$. The concept is illustrated in Figure 3. The authors propose to obtain robust predictions by using the mean confidence of the ensemble members as a combined output. We extend this by calculating a dispersion metric from the individual outputs and using it as a measurement of uncertainty and including it into decision making. The biggest advantage of this method is the fast inference speed as it is n times faster than an implicitly computed ensemble of the same size using dropout as the necessary modifications to the model architecture do not affect the overall number of computations in a meaningful way. However, in contrast to dropout, we cannot increase the ensemble to any desired size as the maximum possible ensemble size is bound by the model capacity and has to be determined empirically.

3. EXPERIMENTS

In this section, we evaluate the capabilities of our concept using the three methods presented in section 2.3. We performed experiments to answer the following questions:

1. Which method produces large and reliable certainty groups?
2. How does the target accuracy for Certainty Groups affect their size?
3. Do different methods detect the same instances in Certainty Groups?

3.1. Experimental Setup

Our approach is evaluated using two data sets originating from PHM settings: the FordA data set obtained from (Bagnall, 2022) and the AI4I predictive maintenance dataset (Matzka, 2020) obtained from (Dua & Graff, 2017). The FordA dataset consists of 3601 train and 1320 test instances, each consisting of 500 sensor features. The goal is to classify whether the sensor measurement indicates a problem in an automo-

tive subsystem or not. The AI4I Dataset is from synthetic origin and aims to reflect real predictive maintenance data. It consists of 10 000 data points with 6 Features (5 numeric, 1 categorical) and a label. Failures can result from five independent failure modes. As this dataset has a very imbalanced class distribution, we used SMOTE (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) to overcome this issue during training. The experiments were performed on a system with a 8-Core CPU, a NVIDIA RTX 2080 Super GPU and 32GB RAM. For all experiments the same fixed random seed was chosen to ensure reproducibility. The code for these experiments is provided in (CertaintyGroups, 2022).

For each data set, we first developed two architectures, one for estimating uncertainty using MC dropout and one with a MIMO configuration. The dropout model was additionally used for the baseline approach in which the raw confidences are used as a certainty criterion. For these experiments, dropout was not active during inference. We then trained and evaluated both model architectures while performing hyperparameter optimization with respect to the model parameters (hidden dimensions, configurations of the convolutional layers) and training parameters (number of epochs, batch size, learning rate). Note that at this stage, the additional hyperparameter T_s is not yet optimized because it is not yet used. After having found a set of well-performing model hyperparameters, we started to optimize T_s by repeatedly computing certainty groups until we were satisfied with the predictive accuracy in the certainty groups. We then evaluated both the predictive performance and the certainty group behavior on a held-out test set.

The model architectures for each data set are based on similar principles. For both approaches, we tested fully connected and convolutional architectures. The networks are 4 to 7 layers deep. The CNN approach is built like a LeNet-Style CNN (LeCun, Bottou, Bengio, & Haffner, 1998) with a feature extraction module built from convolutional layers at the beginning of the network and a final classification stage at the end of the network. In contrast to image classification, we use 1D-Convolutions as our problems only consist of 1D Data (e.g. sensor readings). Depending on the data set, minor changes to the internal dimensions of the layers were necessary. The main differences between the architectures for the dropout and the MIMO approach are dropout layers and reshape/concatenation operations before the first layer. These differences can be easily built into the base architectures. During the development of the model architectures, we followed the principles and goals stated in section 1. We therefore aimed to use architectures that are well manageable, which is the reason why we have refrained from using larger architectures. Figures 4 and 5 illustrate implementations for both approaches. Table 1 lists the architecture type per data set we chose for the evaluation. It can be seen that the fully connected architectures were used for the problem with fewer

Table 1. Used architecture types per dataset and approach for uncertainty estimation.

Dataset	Dropout Model	MIMO Model
AI4I	Fully Connected	Fully Connected
FordA	Convolutional	Convolutional

input features. Havasi et al. suggested 2 or 4 as an optimal number of subnetworks in a MIMO configuration. In our experiments with less complex datasets, we found that 16 (AI4I) and 32 (FordA) subnetworks lead to optimal results.

3.2. Results

As a first step and to be able to judge the baseline model performance, we evaluated the predictive accuracy of the different models over the whole test set without partitioning them into normal and certainty group. As the AI4I dataset is not perfectly balanced, we additionally used *balanced accuracy (BA)* as a metric. Balanced accuracy is especially useful with imbalanced datasets in which a zero-rule classifier can lead to very high accuracy scores. In case of binary classification, it is defined as following:

$$BA = \frac{Sensitivity + Specificity}{2} \quad (8)$$

The accuracy and balanced accuracy scores for each dataset and architecture are shown in Table 2. For the FordA datasets we were able to achieve good accuracies well above 90%. On the AI4I dataset we achieved 84.0% and 80.74% accuracy. It was the hardest to train and showed fragile training behaviour, which may be caused by its low number of features. Note that we did not use the ensembling capabilities of the dropout and MIMO models during the evaluation of the predictive performance as this was not our main goal.

3.2.1. The 100% accuracy requirement for CGs

This section is dedicated to answer the first research question: *Which method produces large and reliable certainty groups?* For these experiments, we optimized the hyperparameter T_s for the dropout and the MIMO architectures w.r.t. maximum predictive accuracy in the certainty group. To accomplish this goal, we chose values for T_s so that the certainty group accuracy on a validation set was 100%. The chosen values for T_s were then evaluated on a dedicated test set. The results of these experiments can be found in tables 3 and 4. A certainty group size of $X\%$ means that $X\%$ of all instances-prediction pairs fulfill the certainty criterion and thus can be seen as “safe”. We do not give the balanced accuracy scores in this case, as this metric loses its expressiveness when very few mispredictions (1-2 wrong predictions versus hundreds of correct) occur. Our concept was able to extract 26.30% to 50.61% of all predictions, with them having an accuracy

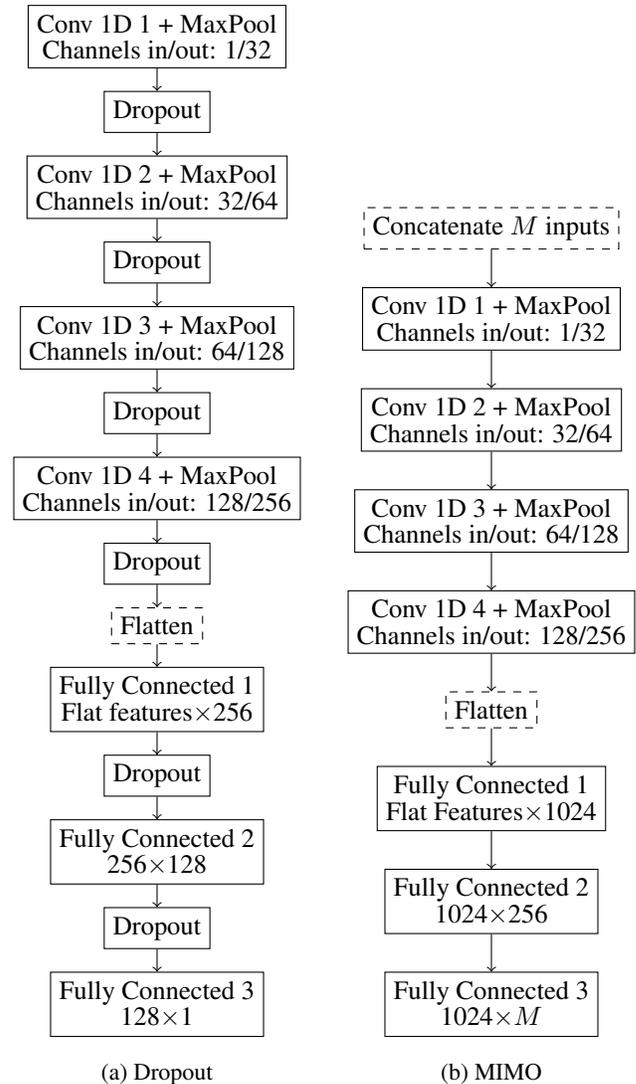


Figure 4. CNN based architectures for both approaches.

Table 2. Accuracies of the models used for the Certainty Group Computation

Dataset	Dropout Model	MIMO Model
AI4I	Acc.: 84.0%; BA 87.06%	Acc.: 80.74%; BA: 79.13%
FordA	Acc.: 91.66%; BA: 91.73%	Acc.: 93.06%; BA: 93.02%

Table 3. Certainty group sizes on the test set when tuned to 100% accuracy on the validation set. If no accuracy score is presented there were no mispredictions

Dataset	Dropout Model	MIMO Model	Confidence
AI4I	26.3%	47.1% with Acc. 99.78%	42.2 %
FordA	45.30% with Acc. 99.66%	50.61% with Acc. 99.25%	54.69% with Acc. 98.61%

score of 99.25% to 100.0%. It can be seen that the targeted accuracy of 100% in the certainty group is not always achieved. Especially the MIMO models were not able to translate this accuracy from the validation set to the test set. If perfect predictions in the certainty group are a strict requirement, a more conservatively chosen value for T_s may be necessary. On all data sets it can be seen that the corresponding normal groups have a clearly worse predictive accuracy. For comparison with the dropout and MIMO approaches, we used the raw confidences of the dropout architecture to compute certainty groups by only accepting predictions with a confidence ≥ 0.99 or ≤ 0.01 into the certainty group. We chose these values for T_{lower} and T_{upper} because we saw them as a natural choice for viewing predictions as “safe” if no uncertainty estimation strategy is implemented. Using this method, our concept extracted 42.2% and 54.69% of all predictions, with them having an accuracy score of 98.61% and 100.0%. It outperformed the dropout approach on both datasets and was able to outperform the MIMO approach on the FordA dataset, however with a slightly lower accuracy. In section 4.1 we take a closer look at this result.

3.2.2. The 98% accuracy requirement for CGs

To demonstrate the flexibility of our approach, we performed additional experiments in which we tuned T_s on an already trained model for ca. 98.0%-98.5% accuracy in the certainty group. For the confidence approach, we tuned the margin thresholds T_{lower} and T_{upper} until the desired accuracy was reached. This was done to answer the question of *how does the target accuracy for Certainty Groups affect their size*. It is especially interesting for use cases that don't require a maximum amount of certainty. The results of these experiments can be found in Table 5. It can be seen that the certainty groups contain 50.6% to 83.40% of all instances. These predictions have accuracy scores ranging from 96.7% up to 98.33%. With more instances and false predictions in the certainty group, we added balanced accuracy scores for these experiments to put the accuracy scores into perspective. This is important to interpret the MIMO model's performance

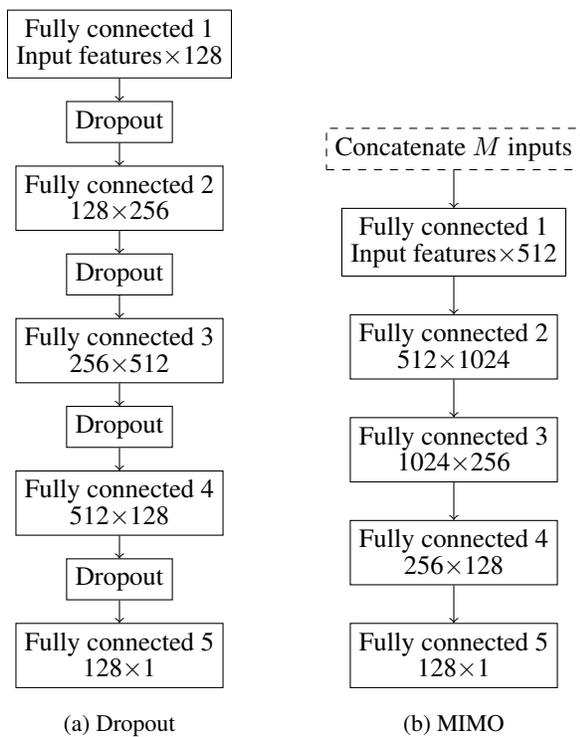


Figure 5. Fully connected architectures for both approaches.

Table 4. Corresponding Sizes and test set accuracies of the normal groups to the certainty groups from Table 3.

Dataset	Dropout Model	MIMO Model	Confidence
AI4I	73.7%, Acc.: 76.11%	52.8%, Acc. 70.88%	57.8%, Acc.: 72.31%
FordA	54.70%, Acc. 87.81%	49.39%, Acc. 88.19%	45.3%, Acc.: 83.27%

Table 5. Certainty Group Sizes on the test set when tuned to ca. 98% Accuracy on the validation set.

Dataset	Dropout Model	MIMO Model	Confidence
AI4I	50.60%, Acc. 98.22%, BA 99.09%	56.49%, Acc. 98.93%, BA 74.73%	78.0%, Acc. 98.33%, BA 95.37%
FordA	75.60%, Acc. 97.99%, BA 97.99%	83.40%, Acc. 96.73%, BA 96.73%	68.78%, Acc. 97.57%, BA 97.44%

AI4I data set In this case, the approach was able to extract large portions of the predictions into the certainty group, but showed flawed predictive performance w.r.t one data set class. In these experiments the naive confidence approach behaved slightly different compared to the experiments in which we aimed towards 100.0% accuracy in the certainty group. While it previously performed best on the FordA data set, it now performs worst. On the AI4I data set however the confidence approach was able to outperform the other approaches by a significant margin. The dropout approach was always able to keep both the balanced and the standard accuracy score on a similar level and performed as desired or even better.

3.2.3. Evaluation of intersection

We performed the following experiments because we were interested in the agreement of the different approaches, i.e. *do different methods detect the same instances in Certainty Groups?* That would be an indication that some instances are inherently harder or easier to predict, perhaps due to the available measurements, quality of data, and non-ambiguity. A real-world example would be the difference in “hardness” between a very sharp, high-resolution image and a blurry, very low-resolution image. We assumed that if an instance is placed into the certainty groups by *each* approach, it really must be a certain prediction and an easy-to-classify instance. Another perspective on this assumption is the difference between *aleatoric* and *epistemic* uncertainty. While epistemic uncertainty is a result of insufficient model capability, aleatoric uncertainty arises directly from randomness and/or variabilities of the underlying data source (Murphy, 2022). If instances are accepted to the certainty group by multiple approaches, they have a low aleatoric uncertainty. For quantifying the agreement between the different approaches, we developed a metric that measures how many of the instances in the smallest certainty group are contained in the common certainty group consisting of instances that have been accepted by every approach. The metric measures the ratio of the overlap of all certainty groups and the size of the smallest certainty group. We assumed that it’s the best case if all accepted instances of a weaker approach (w.r.t. the maximum

possible certainty group size) are part of the overlap of all certainty groups. We call this metric *intersection* and it is defined as following:

$$\frac{Overlap}{|\min(CG_{Dropout}, CG_{MIMO}, CG_{Confidence})|} \quad (9)$$

The described best case scenario leads to an intersection score of 100%. The intersection results for all three datasets are shown in Table 6. The results were calculated using the results of the 100% target experiments. It can be seen that in all two datasets the smallest certainty group is contained to at least 75% in the overlap group. This suggests that the combination of established approaches for uncertainty estimation and the concept of certainty groups is able to extract instances from an underlying dataset that are inherently easier and clearer to predict than other instances.

4. DISCUSSION

4.1. Analysis

A conclusive assessment of all the approaches evaluated is difficult to make as the approaches show different strengths and weaknesses. If we only consider the sizes of the certainty groups, the MIMO approach was the best performing although it is outperformed by the confidence approach on one data set both in the 100% and 98% target experiments. Despite showing the greatest potential in our initial experiments, the MIMO approach had some problems when taking all results into account. It was the only approach that was not able to produce certainty groups with no mispredictions on a held-out test set when tuned on a validation set. Additionally, in the 98% target certainty group the balanced accuracy was not good in one of the two datasets. Its greatest asset in our experiment was the extremely fast execution speed as it only requires one inference pass and its good performance in the 100% target certainty group.

The fact that the confidence approach worked so well was surprising for us. We have therefore carried out further analysis with the framework introduced in (Küppers, Kronenberger, Shantia, & Haselhoff, 2020) to get a clearer picture of the

Table 6. Certainty Group sizes for each approach and dataset and the resulting intersection scores. The Certainty Group sizes are the same as in Table 3 since the same models and certainty criteria were used.

Dataset	Dropout CG	MIMO CG	Confidence CG	Intersection
AI4I	26.30%	47.10%	42.20%	76.42%
FordA	45.30%	50.61%	54.69%	75.91%

actual model calibrations. All models we used for the confidence approach were significantly miscalibrated. This finding collided with our hypothesis that a model has to be well calibrated in order to work well with this approach. It even appears that miscalibration in the margins used for the certainty criteria is beneficial for this approach as it helps to avoid mispredictions when the used margins are large (e.g. confidence >0.7 or <0.3). If a model is perfectly calibrated, predictions at confidence 0.75 should be 75% accurate. If the model is massively miscalibrated in this region (e.g. confidence 0.75 yields 100% accuracy), this helps in the setting of certainty groups. Overall, this approach seems to be very dependent on the model calibration, which poses risks for real-world applications.

The MC-dropout approach was the most consistent performing. Although it was not able to consistently outperform the other two approaches w.r.t. certainty group sizes, it always worked as expected across all datasets and experiments. This is underlined in the 98% target scenario, in which the dropout approach outperformed both other methods w.r.t. accuracy and balanced accuracy. On the AI4I dataset, its balanced accuracy score was clearly better, while on the FordA dataset it outperformed the accuracy score of the other two methods. However, a major disadvantage of this approach is the slow execution speed that was clearly noticeable in the experiments.

In summary, the confidence approach entails risks for real world operation as the model calibration plays a massive role in the creation of the certainty groups. The MIMO approach is able to generate relatively large certainty groups but works best when $>99.5\%$ accuracy is targeted in the certainty group. The dropout approach is overall very consistent but suffers from slow execution speed.

4.2. Possible applications

The results have made us optimistic that our approach is suitable for several applications in PHM. It can be applied to use cases in which a clear and concise decision is needed for an ML-system to add value. The results show that our approach is able to extract a subset of instances with highly accurate predictions. This can enable a transition towards increased automation of data based decisions in industrial applications since not all decisions are put into the hands of an ML-based system. It thus represents a compromise between conservative and progressive data analytics approaches.

The experiments show that our approach is able to extract subsets of the dataset with a higher predictive accuracy compared to the whole dataset. Depending on the setting, our approach is also capable of making models with mediocre predictive quality at least partially useful (e.g. MIMO on AI4I with the 100% target). In situations where, for example, the data source or the available hardware does not allow a more powerful model, this could prove useful. This could enable multi-level architectures in which data points “flow” through multiple levels of machine learning models. A conceivable use case is the distributed IT architecture in manufacturing plants in which very small computational units (the so called “edge”) represent the lowest level of devices and high performance computer systems either in the cloud or in local data centers represent the highest level of computational capability. Our approach could allow very small but not well performing models to be deployed as close to the production line as possible. In this stage, predictions in the certainty group are considered “clear” and safe to proceed without further supervision. The instances in the normal group can then flow towards higher levels of computational capability in which the same process is repeated with more capable ML methods. At the end stands either a completely automatic decision or human supervision for very hard cases. This concept can be interpreted as an iterative reduction of *epistemic uncertainty* as increasingly capable models are applied to the underlying problem.

4.3. Limitations and future work

Despite showing potential for several use cases (as described in section 4.2), our approach in its current state suffers from a couple of limitations, which yield potential for further research of the presented concept.

Currently, our approach does only work with binary classification. In initial experiments on multiclass problems conducted at an early stage of research, we found that the expressiveness of the predictive standard deviation is lower than in a binary setting. We plan to investigate other dispersion metrics, perhaps information-theoretic ones based on entropy or KL-divergence. Using our collected knowledge, an adoption of certainty groups for multiclass classification would be the next logical step. A variation of the problem in the form of one versus all would be conceivable as in this case a binary aspect would still be present.

Despite the possibility of tuning T_s such that the accuracy on

a validation set is 100%, our approach cannot guarantee perfect predictions in a real world application. As shown in the results, it is still possible that all ensemble members perform wrong predictions with a very similar confidence that leads to a containment in the certainty group. Additionally, our approach does not take distribution shifts into account. In a real world application in which distribution shift naturally occurs (e.g. through wear and tear of multiple machine components), some type of out-of-distribution (OOD) detection would be necessary to prevent false predictions in the certainty group.

So far we have tuned T_s manually to result in a desired validation accuracy for the certainty group. An automatic solution for this task would be another next logical step, perhaps using Bayesian optimization (Mockus, 2012) or other optimization techniques. This could also involve requirements for the certainty group that are more complex than only accuracy requirements. An optimization with regards to very specific predictive behavior could be implemented by more complex certainty and optimization criteria. A conceivable example is the avoidance of false negative predictions in a predictive quality setting. In this case, a false negative prediction (defect not detected) is much more problematic than a false positive prediction. An optimizing solution could look for a certainty criterion on the uncertainty measure ζ_m that matches the target metrics for the certainty group.

While we only considered neural networks in this work, our approach is not limited to them. An in-depth comparison of the certainty group behavior of the used neural networks to Bayesian approaches (e.g. Gaussian processes or Laplace approximations) as hinted in Sect. 2.1 would be very interesting. So far we have evaluated our approach only with relatively small architectures. We have also not investigated the impact of the actual model architecture (e.g. convolutional, fully connected, recurrent) onto the quality and size of the certainty groups. Therefore, a direct comparison between the smaller architectures and architecture types we used to significantly larger architectures and different architecture types would add to the understanding of the presented concept. As the confidence approach showed serious potential, the application of calibration techniques like in (Guo et al., 2017) or (Tomani & Buettner, 2021) could result in interesting findings and are worth investigating.

5. CONCLUSION

In this paper, we introduced the concept of certainty groups that divides the predictions of neural networks into a normal group and the certainty group. Through an additional hyperparameter, the quality of the predictions in the certainty group can be controlled, which can enable interesting applications in manufacturing applications relevant to prognostics and health management. We evaluated three approaches for computing certainty groups, with an approach based on

MC dropout performing most consistent despite not generating the largest certainty groups. We further showed that the three approaches accept very similar instances into the certainty group. Our approach is characterized by its simplicity and practical motivation as it combines advanced uncertainty estimation techniques with a clear certainty criterion.

REFERENCES

- Bagnall, A. (2022). *Time Series Classification*. Retrieved from <http://www.timeseriesclassification.com/>
- Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., ... Goodman, N. D. (2018). Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*.
- CertaintyGroups. (2022). *Certainty Groups Code on Anonymous GitHub*. Retrieved from <https://anonymous.4open.science/r/CertaintyGroups>
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321–357.
- Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., & Hennig, P. (2021). Laplace redux - effortless bayesian deep learning. *CoRR*, abs/2106.14806. Retrieved from <https://arxiv.org/abs/2106.14806>
- Dua, D., & Graff, C. (2017). *UCI machine learning repository*. Retrieved from <http://archive.ics.uci.edu/ml>
- Frankle, J., & Carbin, M. (2018). The lottery ticket hypothesis: Training pruned neural networks. *CoRR*, abs/1803.03635. Retrieved from <http://arxiv.org/abs/1803.03635>
- Gal, Y., & Ghahramani, Z. (2016). *Dropout as a bayesian approximation: Representing model uncertainty in deep learning*.
- Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International conference on machine learning* (pp. 1321–1330).
- Harshavardhanan, S., & Nene, M. J. (2020). Capturing and modeling uncertainty in prognostics and health management using machine learning. In *2020 5th international conference on communication and electronics systems (icces)* (pp. 1235–1241).
- Havasi, M., Jenatton, R., Fort, S., Liu, J. Z., Snoek, J., Lakshminarayanan, B., ... Tran, D. (2020). Training independent subnetworks for robust prediction. *CoRR*, abs/2010.06610. Retrieved from <https://arxiv.org/abs/2010.06610>
- Ismail, M., Mostafa, N. A., & El-assal, A. (2021). Quality monitoring in multistage manufacturing systems by us-

- ing machine learning techniques. *Journal of Intelligent Manufacturing*, 1–16.
- Küppers, F., Kronenberger, J., Shantia, A., & Haselhoff, A. (2020, June). Multivariate confidence calibration for object detection. In *The IEEE/CVF conference on computer vision and pattern recognition (cvpr) workshops*.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
- Lee, S., Purushwalkam, S., Cogswell, M., Crandall, D. J., & Batra, D. (2015). Why M heads are better than one: Training a diverse ensemble of deep networks. *CoRR, abs/1511.06314*. Retrieved from <http://arxiv.org/abs/1511.06314>
- Matzka, S. (2020). Explainable artificial intelligence for predictive maintenance applications. In *2020 third international conference on artificial intelligence for industries (ai4i)* (pp. 69–74).
- Mockus, J. (2012). *Bayesian approach to global optimization: theory and applications* (Vol. 37). Springer Science & Business Media.
- Molchanov, P., Tyree, S., Karras, T., Aila, T., & Kautz, J. (2016). Pruning convolutional neural networks for resource efficient transfer learning. *CoRR, abs/1611.06440*. Retrieved from <http://arxiv.org/abs/1611.06440>
- Murphy, K. P. (2022). *Probabilistic machine learning: An introduction*. MIT Press. Retrieved from probml.ai
- Naeini, M. P., Cooper, G., & Hauskrecht, M. (2015). Obtaining well calibrated probabilities using bayesian binning. In *Twenty-ninth aai conference on artificial intelligence*.
- Phan, D., Pradhan, N., & Jankowiak, M. (2019). Composable effects for flexible and accelerated probabilistic programming in numpyro. *arXiv preprint arXiv:1912.11554*.
- Salvatier, J., Wiecki, T. V., & Fonnesbeck, C. (2016). Probabilistic programming in python using pymc3. *PeerJ Computer Science*, 2, e55.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929–1958.
- Tehrani, N., Arora, N. S., Li, Y. L., Shah, K. D., Noursi, D., Tingley, M., ... others (2020). Bean machine: A declarative probabilistic programming language for efficient programmable inference. In *International conference on probabilistic graphical models*.
- Tomani, C., & Buettner, F. (2021). Towards trustworthy predictions from deep neural networks with fast adversarial calibration. In *Thirty-fifth aai conference on artificial intelligence* (Vol. 3).
- Weiss, M., & Tonella, P. (2021). Uncertainty-wizard: Fast and user-friendly neural network uncertainty quantification. *CoRR, abs/2101.00982*. Retrieved from <https://arxiv.org/abs/2101.00982>
- Wen, Y., Tran, D., & Ba, J. (2020a). Batchensemble: An alternative approach to efficient ensemble and lifelong learning. *CoRR, abs/2002.06715*. Retrieved from <https://arxiv.org/abs/2002.06715>
- Wen, Y., Tran, D., & Ba, J. (2020b). Batchensemble: An alternative approach to efficient ensemble and lifelong learning. *CoRR, abs/2002.06715*. Retrieved from <https://arxiv.org/abs/2002.06715>
- Zhang, L., Lin, J., Liu, B., Zhang, Z., Yan, X., & Wei, M. (2019). A review on deep learning applications in prognostics and health management. *Ieee Access*, 7, 162415–162438.
- Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2016). Deep learning and its applications to machine health monitoring: A survey. *CoRR, abs/1612.07640*. Retrieved from <http://arxiv.org/abs/1612.07640>
- Zhu, M., & Gupta, S. (2017). To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*.

BIOGRAPHIES

Lukas Lodes grew up in Bavaria, Germany. He studied computer science at the University of Augsburg (Germany), where he obtained his Bachelor’s and Master’s degree. His master thesis focused on the application of graphical neural networks for the prediction of CFRP manufacturing results. He is currently working as a research assistant and PhD student under the supervision of Alexander Schiendorfer at THI, the Technical University of Applied Sciences Ingolstadt.

Alexander Schiendorfer grew up in Upper Austria where he also obtained his Bachelor’s degree in Software Engineering at the UAUaS in Hagenberg. He then went on to get a joint Master’s degree in Software Engineering from the University of Augsburg, TUM (Technical University of Munich), and LMU (the University of Munich). After receiving his doctoral degree in Computer Science at the University of Augsburg in 2019 and two years as a postdoctoral scholar, he accepted a professorship at Technische Hochschule Ingolstadt (THI). His research interests lie in the application of combinatorial optimization and machine learning to the manufacturing domain – in particular constraint programming, probabilistic machine learning, and reinforcement learning. He is a member of the Association for Constraint Programming (ACP) and won the University of Augsburg doctoral dissertation award in 2019.

Processing of Condition Monitoring Annotations with BERT and Technical Language Substitution: A Case Study

Karl Löwenmark¹, Cees Taal², Joakim Nivre³, Marcus Liwicki¹ and Fredrik Sandin¹

¹ *Embedded Intelligent Systems Laboratory (EISLAB), Luleå University of Technology, 971 87 Luleå, Sweden, karl.ekstrom@ltu.se*

² *SKF Research & Technology Development, Meidoornkade 14, 3992 AE Houten, P.O. Box 2350, 3430 DT Nieuwegein, The Netherlands*

³ *RISE Research Institutes of Sweden, Isaffjordsgatan 22, 164 40 Kista, Sweden, P.O. Box 857, 501 15 Borås, Sweden*

ABSTRACT

Annotations in condition monitoring systems contain information regarding asset history and fault characteristics in the form of unstructured text that could, if unlocked, be used for intelligent fault diagnosis. However, processing these annotations with pre-trained natural language models such as BERT is problematic due to out-of-vocabulary (OOV) technical terms, resulting in inaccurate language embeddings. Here we investigate the effect of OOV technical terms on BERT and SentenceBERT embeddings by substituting technical terms with natural language descriptions. The embeddings were computed for each annotation in a pre-processed corpus, with and without substitution. The K-Means clustering score was calculated on sentence embeddings, and a Long Short-Term Memory (LSTM) network was trained on word embeddings with the objective to recreate the output from a keyword-based annotation classifier. The K-Means score for SentenceBERT annotation embeddings improved by 40% at seven clusters by technical language substitution, and the labelling capacity of the BERT-LSTM model was improved from 88.3 to 94.2%. These results indicate that the substitution of OOV technical terms can improve the representation accuracy of the embeddings of the pre-trained BERT and SentenceBERT models, and that pre-trained language models can be used to process technical language.

1. INTRODUCTION

Condition monitoring is vital in the process industry to ensure safe production with minimal wastage and early stops. Monitoring is done by human analysts, assisted by a computerized

maintenance management system to estimate the state of industry components and make maintenance decisions. Upon fault detection, maintenance decision or maintenance activity (e.g. component replaced), the outcome is sometimes stored as unstructured text in maintenance work orders (MWOs) and fault diagnosis annotations. These MWOs and annotations contain valuable condition monitoring process information that could be used for data analysis purposes, if the knowledge embedded in the text could be unlocked and integrated into a computerised system. Processing this text would thus improve feedback between analysts and systems, and facilitate learning-based implementations using process-specific technical knowledge.

Transformer-based (Vaswani et al., 2017) language models such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2019) have been successfully used in many natural language processing (NLP) domains, but are not yet as widely implemented in the technical language domain. Word representations in BERT-based models are typically computed as functions of entire input sequences (Peters et al., 2017, 2018). The contextual input sequence approach adopted from ELMo (Peters et al., 2018) has many advantages, such as dealing with polysemy – identical words having different meanings in different contexts – which can thus be handled, as the embedding for a specific word is different depending on its context.

Technical Language Processing (TLP) was introduced as a domain-driven approach to NLP in a technical engineering framework (Brundage et al., 2021) to help unlock the knowledge represented in technical text. The potential for implementations of embedding algorithms on technical language was investigated by Nandyala et al. (2021), where the challenges of technical language word representations are further discussed, and five suggestions of word representation sys-

Karl Löwenmark et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

tems are provided: TF-IDF, pre-trained Word2Vec (Bahdanau et al., 2015), pre-trained GloVe (Pennington et al., 2014), custom-trained Word2Vec, and pre-trained BERT (Devlin et al., 2019). The authors presented filtered word clusters and word and sentence similarities using these models with a TLP pipeline on 5,485 work orders for 5 excavators (Hodkiewicz et al., 2017).¹ Cadavid et al. (2020) applied CamemBERT, a French version of BERT, to estimate the criticality and duration of maintenance problems from operator descriptions in a French manufacturing company dataset.

Due to the success of the BERT architecture there are many derivative models. A popular model is RoBERTa (Liu et al., 2019), which is a retrained BERT model with different hyperparameters and a slightly different pre-training objective. Neither BERT nor RoBERTa are explicitly trained for sentence embeddings however, as both produce embeddings at a word-level. Typically, either the embedding of the last token (CLS) in the sequence, the mean of all embeddings in the sequence, or the max embedding, are used. SentenceBERT (Reimers & Gurevych, 2019) is a BERT-based model fine-tuned for sentence similarity with siamese and triplet networks (Schroff et al., 2015), with output vectors being pooled to a fixed-size sentence embedding using either the output of the CLS token, the mean of each output embedding vector, or a max-over-time computation of the output vectors. Using SentenceBERT, it is possible to represent the semantics of sentences and complete annotations more accurately than with BERT, which performs better on word-level representations.

Another solution for language representation is a keyword-based labelling system taxonomy. Ottermo et al. (2021) worked with domain experts to build a taxonomy from 80 annotations for classification of failure events in oil and gas valves in Norway, creating named entities from technical text to symbolise fault cases. A keyword-based system can perform well on a limited data set without requiring labels, but requires human engineering and scales poorly due to the limits of pre-defined conditional statements and inherent complexity in language.

Embeddings from pre-trained language models scale better, but do not generalise well to completely new domains and require large data sets to train. Evaluating the performance of technical word embeddings is also challenging, as most NLP evaluations rely on human-labelled downstream tasks such as GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019), but no such datasets exist for technical language. However, a well-developed technical keyword-based labelling system could serve as one basis for evaluation, and out of vocabulary (OOV) technical terms can be substituted by adding technical taxonomies based on natural language, effectively combining the benefits of both systems. Thus,

¹Also known as the Prognostics Data Library.

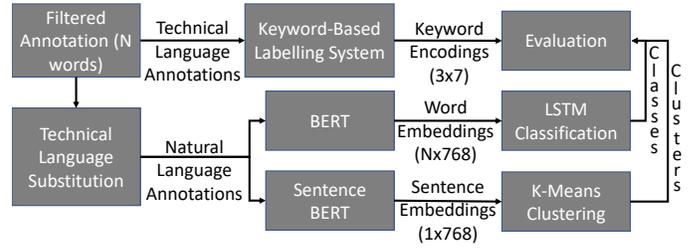


Figure 1. Overview of technical language processing steps, including technical language substitution and evaluation.

methods for the integration of OOV technical terms need to be developed and evaluated to further the integration of NLP models into the TLP framework.

Here we investigate the effect of substituting technical terms with natural language descriptions or synonyms on the word and sentence representations generated by a pre-trained language model. In particular, we examine the technical language representations of BERT and SentenceBERT with and without technical language substitution through K-Means clustering, t-SNE (Hinton & Roweis, 2002) analysis and automatic labelling with a keyword-based system. The dataset used consists of real MWOs and annotations from vibration sensor condition monitoring of two large paper manufacturing plants in northern Sweden.

2. METHOD

Figure 1 gives an overview of the methodology, from annotations to clusters and evaluation. Filtered annotations are used as inputs to the technical language substitution block, presented in Section 2.1, where important technical words are identified and replaced with in-vocabulary language synonyms or descriptions. A pre-processed technical language annotation consisting of N words is fed directly into the keyword-based labelling system, outlined in Section 2.2, which produces labels that can serve as valuable evaluation insights for the unsupervised systems. The effect of the substitution on language model representations is evaluated through a supervised classification task and an unsupervised clustering task, further described in Section 2.3. Finally, the corpus from which annotations were filtered is described in Section 2.4.

2.1. Technical Language Substitution

The purpose of technical language substitution is to use human knowledge and language understanding to facilitate a training-free transfer to new domains where training-data is limited. Technical terms that are substituted are critically important for the condition monitoring-related semantics of the annotations, such as fault class, which directly guides further fault diagnosis. Prior to substitution, these terms are input as tokens that likely do not capture the semantics of the term, but post substitution, these terms are ideally transformed to accurate representations of the term meaning.

Analysing the vocabulary of a language model can be done by investigating its tokenizer, how it represents word strings with sparse vectors prior to embedding them. Older word vector representations tended to have a fixed vocabulary, being capable only of representing words seen during training, while all out-of-vocabulary (OOV) words are represented with the same "unknown" token. Modern language models typically use techniques to maximise the coverage of all words in a corpus with a set amount of n-grams, through for instance subword tokenisation such as byte pair encoding (BPE) (Gage, 1994; Sennrich et al., 2015) or WordPiece (Schuster & Nakajima, 2012; Wu et al., 2016). The base BERT model and some of its derivatives use WordPiece, while the GPT series and for instance RoBERTa use BPE. Both subword tokenisation algorithms work by representing common words with a single token, while uncommon words are represented as a combination of tokens down to the base character level. For instance, using the WordPiece based *bert-base-uncased BertTokenizer* from the HuggingFace library, the word *academic* is tokenised directly to [*academic*], while *academical* becomes [*academic, ##al*]. Thus, instead of having unique representations for every word ending, the endings can become unique tokens attached to the root words. However, the WordPiece splitting does not always produce intelligible tokens, such as for *remanufacture*, which is tokenised as [*re, ##man, ##uf, ##act, ##re*], despite [*manufacture*] also existing as a token in the tokenizer vocabulary. Nonetheless, while some words might be split into unintelligible word pieces, no words are technically outside of the model's vocabulary.

Table 1 shows which technical terms were substituted and their substitution, translated from Swedish to English for reader convenience. Substitution can be done through paraphrasing – rewriting a technical concept with in-vocabulary words while maintaining semantics; synonym substitution – replacing an OOV-term with a semantically similar in-vocabulary term; or abbreviation expansion – substituting an OOV abbreviation with the in-vocabulary words that constitute it. Tests with fewer and with more substitutions were done, and the effect follows a pattern of more frequent and impactful keywords such as *BPFO* and *WO* having larger effect on the clus-

Table 1. Technical terms and the natural language substitutions used.

technical terms	natural language substitution
bpfo	fault in the outer ring
bpfi	damage on the inner ring
sensor (givare)	sensor
wo	work order
looseness	distance that causes instability
dc	drying cylinder
env/envelope	measurement signal
gr	gear
mms	velocity measurement
fs	free side
ds	drive side

ters and k-Means-score, and less frequent or impactful words such as *mms* and *ds* having smaller effect. The final keywords were chosen to be sufficiently numerous to clearly illustrate the different types impact of substitution, but limiting the number as to not obfuscate the impact through unnecessarily extensive input alterations. A more systematic study of the impact of various keyword combinations is a natural next step to further the understanding of the interaction between technical terms and natural language models.

The natural language substitution was designed to be semantically similar to the technical terms, and pass through the tokenizer in a predictable manner, so that no substitution words were chopped up in a way which indicates that BERT has no prior experience of the term. Therefore, the Swedish word *givare*, which means sensor, was for instance just replaced with *sensor*, as SweBERT tokenises *sensor* as one token. The terms *BPFO* and *BPFI* are common fault types in ball-point bearings with distinct condition indicators in signals, but where fault severity is challenging to assess estimate even for expert analysts. Many technical terms are in effect abbreviations, such as *WO* for *work order*, or *dc* for *drying cylinder*. These can, in some cases such as *WO*, be directly substituted for their unabbreviated terms. Other terms, such as *GC* for *guide roller*, can be more difficult to simply expand. The Swedish word for roller, *vals*, is not encoded as one token, but rather as [*val, ##s*], which means whale's or election's. Even if *vals* was encoded as one token, it would likely refer to the more common meaning waltz, the dance, in pre-training rather than a roller. The issue of polysemy, multiple meanings of the same word, is circumvented in contextual language models by using the context as part of the word embedding input, but when neither the context nor the word meaning has been encountered during pre-training, the output embedding is unlikely to be a good representation of the underlying semantics.

Technical language substitution presents an opportunity to merge human knowledge of technical terms with the representational possibilities of natural language models. Unlike language model implementations on general-domain natural language, which can function without significant human interference on input language, this imposes an additional work load on NLP deployment. However, until a technical language model is developed on a massive technical corpus, or natural language models can be accurately fine-tuned on technical data, it can serve as a great addition to the TLP toolbox and for integration into NLP pipelines. Furthermore, defining technical terms in natural language can serve as a spring board for technical language fine tuning, through for instance contrastive learning (Giorgi et al., 2021).

2.2. Keyword-Based Labelling Model

In order to provide additional evaluation methods, a keyword-based labelling system was designed to output fault class and maintenance action labels for each annotation. Keywords were identified by analysing language distributions in collaboration with domain experts. Figure 2 shows the conditional co-occurrence of the eight most common non-stopword words in the dataset, calculated from the probability of a target word (rows) appearing in the dataset given the context word (columns). The co-occurrence illustrates certain properties of the dataset, such as the sparseness between some technical terms, and the co-dependence of others. For instance, the word *written* has a probability of 1 to appear with *WO*, and is thus always used in the same annotation as the word *WO*, which is due to a common phrase in the dataset being *WO written*. However, the word *WO* is also used without *written* as the context in about half of the annotations, as can be seen from the lower probability in the *WO* row. Hence, including *written* in the keyword-based labelling system would accurately capture all instances of annotations explicitly informing *WO written*, but would exclude annotations that implicitly state the same semantics, such as *WO [number] on BPFO*. Therefore, it is more consistent to base the labelling system on *WO* than on *written*.

The low overlap between key terms indicating fault class, which is seen from for instance the lack of co-occurrence between *play*, *sensor* and *BPFO*, indicates that fault annotations often mention at most one fault class. Therefore, a labelling system for fault class detection can be defined by searching for identified fault class keywords. If multiple keywords are found, they can either be concatenated to a new class, or the annotation can be dropped into a corpus with uncertain annotations, where we opted to choose the second option.

The keyword-based labelling system also outputs maintenance actions when applicable, and thus outputs different labels for *sensor replaced*, *WO [work order] written sensor replacement* and *sensor occasionally malfunctioning*. In the first example, it is an annotation indicating that the fault class *sensor* is remedied, so the corresponding signals should be treated as signals without sensor fault indicators. When *WO written* is a part of an annotation, it indicates a fault which has sufficient severity to warrant replacement. Comparatively, a fault class annotation that does not contain either an indication that it has been replaced or that it should be replaced is simply an annotation that the fault is present.

Table 2. Fault classes and maintenance actions in the keyword-based labelling system.

Fault Classes	BPFI, BPFO, Cable, Play, Imbalance, Disturbance, Sensor
Actions	[WO, replace, change], replaced

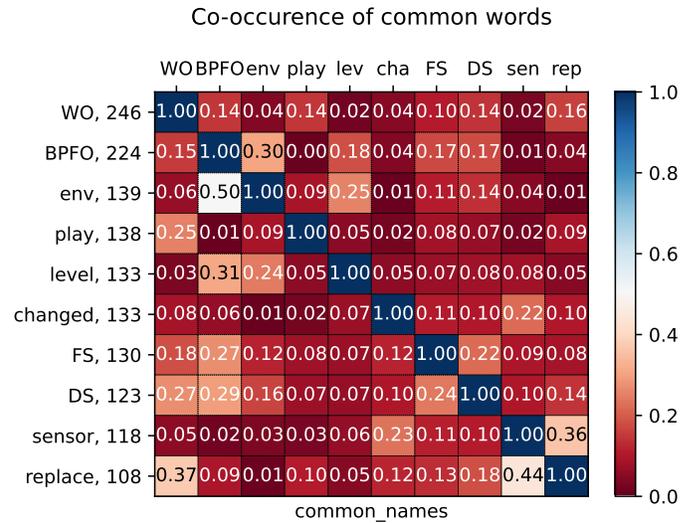


Figure 2. Conditional co-occurrence of common words in the annotations, with stop words removed. The co-occurrence is computed as the probability of the word in the row given the word in the column.

The keywords used in the labelling system are shown in Table 2. If exactly one fault class keyword was detected, the annotation was considered unambiguous and labelled as belonging to that fault class; if zero or multiple class keywords were detected, the annotation was considered ambiguous and put in another dataset. Maintenance actions were defined similarly, but "WO", "replace" and "change" were all projected to the "WO" keyword. Annotations with at most one action keyword type present were considered unambiguous, and annotations with no action keywords were treated as belonging to a "None" class. Only unambiguous annotations were then used in the evaluation step to ensure reliability of the labelling system.

2.3. Evaluation

Three methods for language model performance evaluation were devised. SentenceBERT was used to represent unsupervised language model understanding of complete annotations, while BERT was used together with a Long Short-Term Memory (LSTM) network (Hochreiter & Schmidhuber, 1997) to learn the combined representation through supervision by the keywords-based labelling system.

First, we investigate the embedding properties of SentenceBERT annotation embeddings visually through PCA and t-SNE dimension reduction techniques in an interactive plot where the annotation can be inspected by hovering over the embedded dot. The embedding dimension of 768 is reduced to 50 through PCA, then to two through t-SNE, as it is common practice to avoid applying t-SNE directly on data of dimensions higher than 50. Similar clusters were observed when t-SNE was applied directly on the data, although this required more computation power.

A visual inspection is difficult to quantify in proper evaluation numbers, and prone to subjective bias or focusing on the desired results, but can also serve as a guide to inspect the semantic distributions of annotation embeddings, and offer some insight into what words affect embedding placement in the 2D projection. However, the resulting 2D space can also be investigated quantitatively by clustering the projected embeddings through for instance K-Means, and evaluating the inherent clusterability of the space as well as the words and annotations used in each cluster. If the K-Means score, defined as the average distance between cluster data points and corresponding epicenters, increases (closer to zero) due to a change, then it stands to reason that the vector space is more separable if the score is normalised with regards to the scale of the K-Means space. If the clusters also form based on important technical words indicating fault class or maintenance action, then this improvement in separability is also an improvement in technical language representations. While the classes formed from K-Means might have no inherent overlap with those from the keyword-based labelling system, there is still merit in comparing these to see where the systems agree and disagree.

We also use BERT word embeddings as input to an LSTM network optimised to reproduce the keyword-based labels. The keyword-based labels are certainly not perfect representations of the desired encodings of technical annotations. However, it stands to reason that if the reproducibility of these labels is improved after technical language substitution, in spite of the fact that many technical terms substituted are also directly used as keywords for the labelling system, then the substituted BERT word embeddings are a better representation of the semantic space than the unsubstituted ones.

2.4. Dataset

Table 3 shows corpus properties of the original annotation dataset, a filtered version, a version with only unique annotations and finally annotations marked as clearly defined by the keyword-based system. The original annotations were directly extracted from the condition monitoring dataset, and the filtered annotations were computed by removing digits and special characters from the dataset, which reduces the average annotation length slightly. Duplicate annotations were

Table 3. Properties of differently filtered annotations.

Annotations set	#annotations	μ (annotations length)	σ (annotations length)	#words
Original	1975	6.19	6.50	3008
Filtered	1975	5.71	6.22	1929
Unique	1162	7.13	6.93	1929
Clearly defined	618	6.85	7.43	1111
Unclear	544	7.45	6.29	1334

Table 4. An example of annotations connected to a subasset in a paper machine (with anonymised names, X). This subasset has a higher than average number of annotations due to faults.

Fault	Date	Comment
None	09/20	The lubrication works as it should according to X. This roller is lubricated with oil. The bearing damage is clearly visible in env with many overtones but is at a very low level however it has increase a bit lately. Ground frequency for outer ring in mm/s also seen
BPFO	09/20	BPFO seen on FS. Write WO maybe??? Talked with X and he'll check lubrication.
BPFO	06/20	BPFO Indication DS. Low levels Keep watch.
None	03/20	Roller replaced.
None	12/19	WO written on bearing replacement drive side.
BPFO	12/19	BPFO on drive side. Keep watch.

removed to create a corpus with unique datasets, which has on average longer annotations due to many duplicates being "fault detected" annotations. The dataset was then finally split into clearly and unclearly defined annotations with regards to fault class, as decided by the keyword-based labelling system. The unclearly defined annotations feature on average more words, with on average longer annotations.

The initial corpus consists of 1975 annotations with a total of 3009 unique tokens. Of the 3009 unique tokens, 1286 require wordpiece splitting for BERT to process, as explained in the previous section. The longest annotation is 106 words long, and the shortest is 1 word. The BERT model used as embedder has a maximum input length of 512. With the longest annotation at 106 words, and the tokenised version at most twice as long, all annotations were processable within this output limit. All annotation sentences were transformed from text to embedding space using the BERT tokeniser and the Swedish version of SentenceBERT (Rekathati, 2021).

Table 4 shows examples of annotations associated with a subasset consisting of two sensors, one on the locating (free, FS) and one on the non-locating (drive, DS) side. The first three notes are typical examples of fault detection, maintenance action and maintenance follow up. A BPFO is detected on the drive side, but the severity does not yet warrant a replacement. After 14 days, the analyst decides to write a work order for roller replacement. Three months later, the component is replaced and a follow-up annotation is written. This annotation is of critical importance for the possibility of technical language supervision (Löwenmark et al., 2021), as it indicates where the associated signal data should be treated as healthy again. However, only three months after this replacement a BPFO is spotted again, initially on the non-locating side. Upon further inspection after three months, it now appears primarily on the locating side, which shows the challenging task of analysing signals where faults propagate between sensors. This rapid recurrence of a new fault is unexpected, prompting another annotation describing a more detailed analysis of the component. Since the signal levels are low, the fault has not warranted a replacement even at the end

of the dataset (mid 2021), which showcases the importance of accurate fault severity assessment in minimising unnecessary maintenance actions and material wastage from premature replacements.

3. RESULTS

Figure 3 shows all outputs from the keyword-based labelling system on the filtered corpus, projected onto 2D-representations of SentenceBERT annotation embeddings with and without technical language substitution. The labels were produced as described in Section 2.1, and the 2D visualisation of the embedding space was produced through PCA dimension reduction to 50 dimensions, and t-SNE projection from 50 to 2 dimensions. While the system works identically on both annotations, the visualisation clearly shows that the clusters formed after technical language substitution correlate more with the labels produced by the keyword-based system.

Figure 4 shows 21 dimensional K-Means clustering applied on the same t-SNE projected embedding space as the second half in Figure 3. The labels are formed by searching for the five most common words in annotations belonging to each cluster. Due to BPFO being the most commonly mentioned fault class in the annotation set, it also becomes over-represented in the label set. Comparing to the keyword-based labels from Figure 3, the large sets of BPFO labels (turquoise) has been split into multiple BPFO sets with small nuances in the semantics, such as whether the fault is of low levels or a WO should be written. However, BPFO is clearly over-

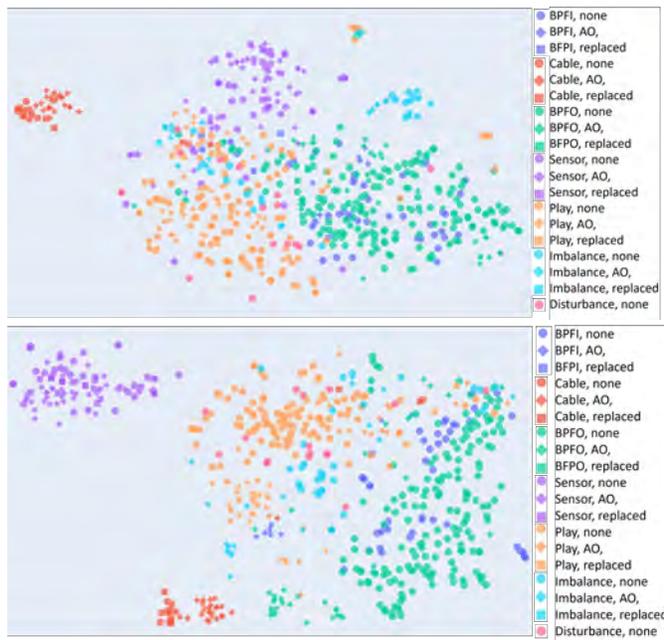


Figure 3. Two dimensional t-SNE transformation of annotation sentence embeddings, labelled by a keyword-based labelling system, without (top) and with (bottom) technical language substitution.

represented in the label set, which likely is due to the high number of BPFO cases, and the wide distribution in the cluster space, as can be seen in the keyword-based labels as well.

The projected embedding space had consistently average K-Means closer to zero for all K larger than 3, and performed significantly better at K larger than 5, as shown in Figure 5. The Figure shows K-Means scores for between 5 and 50 clusters, computed on all unique annotations in the dataset, plotted as a function of number of clusters. As expected, the K-Means score decreases as more clusters can be formed due to the shorter distances between points to their cluster epicenters. The K-Means score is on average 36% more negative before substitution for K larger than 5, and goes below 30% only for K = 11, 12 and 13 in this group.

The effect of technical language substitution was also evaluated by reproducing the output of the keyword-based labelling system from BERT word embedding inputs. The keyword-based labelling system produces annotation labels as outputs, which can be used as supervision signals. To widen the experimental scope, the base Swedish BERT model, KB-BERT (Malmsten et al., 2020), was used on a token level. Thus, an annotation consisting of five tokens resulted in a sequence of five 768 dimensional embeddings. These embeddings were then fed into an LSTM network, whose output was fed into a feed-forward neural network. Compared to SentenceBERT, this effectively allows the model to independently learn the ideal combination of word embeddings into dense representations of annotation semantics.

The LSTM model without technical language substitution obtained an accuracy of 88.3% on a randomly sampled test set when trained for 100 epochs, using the model with the highest validation set accuracy for testing, while the model with technical language substitution obtained an accuracy of 94.2% when trained with an identical set up, for a 5.9% difference. The number of erroneous predictions thus decreased from 11.7% to 5.8% for an error reduction of 50%.

Figure 6 shows an example of confusion matrices from the LSTM systems, where the bottom matrix is computed with substituted input. Due to random sampling of train, validation and test annotations, the number of annotations of each class can differ in each run, which is why multiple trials is important for reliability of the results. While the substituted input is clearly better suited for downstream NLP tasks, both models have the worst accuracy for *Play*, which likely appears in similar context as other faults more than any other fault class. Looking at Figure 3, some *Play* labels are far away from the main cluster in the t-SNE space, so it stands to reason that for the erroneous predictions the embeddings were poor indicators of the *Play* fault class.

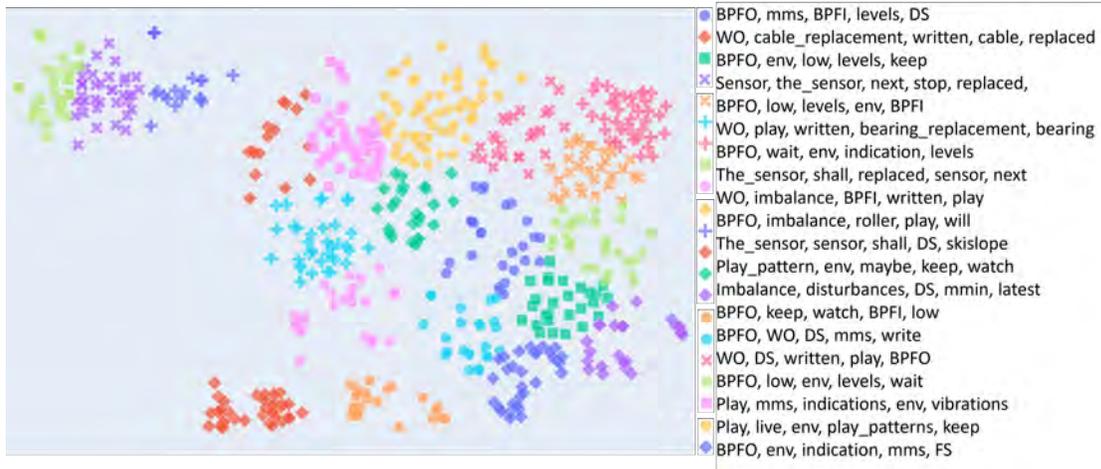


Figure 4. Two dimensional t-SNE transformation of annotation sentence embeddings with technical language substitution, classified with K-Means and labelled with the five most common words per cluster.

4. DISCUSSION

The positive impact of technical language substitution was evaluated qualitatively in Figure 3 and quantitatively through the K-Means score in Figure 5, as well as the LSTM output reconstruction shown in 6. These results are proof of concept that substituting out-of-vocabulary words can improve the language model performance on other language domains. Figure 4 illustrates how well a K-Means algorithm can cluster annotation semantics based on SentenceBERT. Analysing the clusters quantitatively there are few instances where multiple fault classes are among the top five words, with the exception being BPFI which appears after BPFO and lacks its own clusters. The clusters also tend to form around words either indicating low severity, high severity or component replacement, but in a few cases contain both *replace* and *replaced*, words that indicate significant differences in the condition indicators. To deploy a language model on an annotation set, a human-in-the-loop approach where these differences are instructed to the TLP model might be necessary. Nonetheless, the results indicate that substitution significantly improves

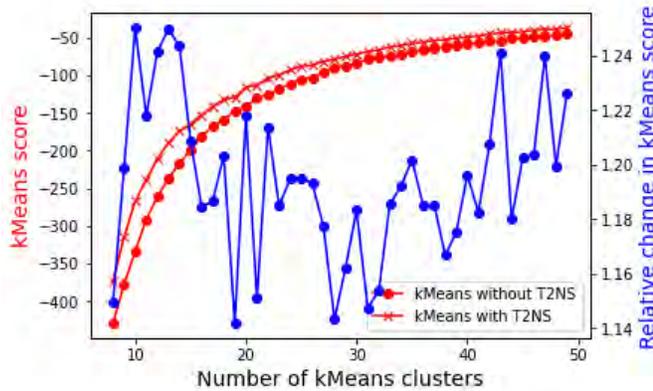


Figure 5. The K-Means score of K-Means clusters formed from SentenceBERT embeddings of unsubstituted (without T2NS) and substituted (with) annotations for a varying number of clusters, the relative improvement calculated as the old K-Means score divided by the new score.

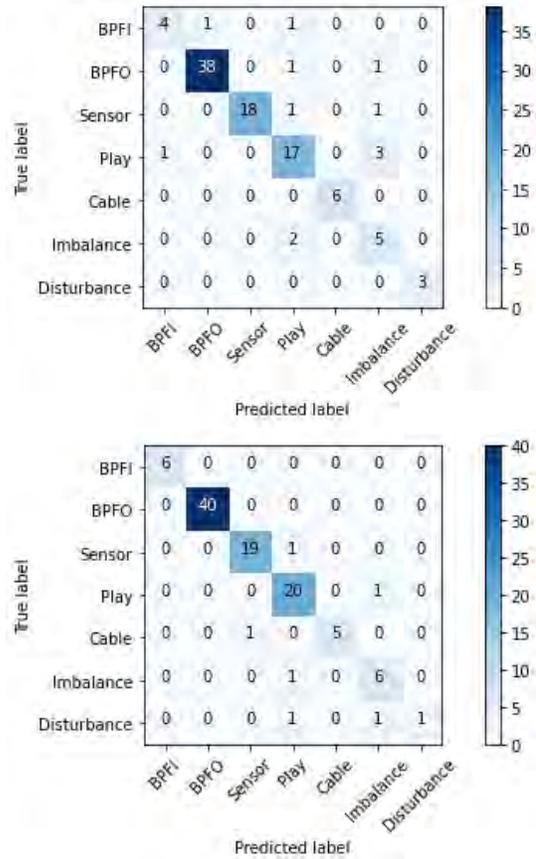


Figure 6. Confusion matrices from an LSTM with fully connected layers reconstructing the keyword-based labelling system classification without (top) and with (bottom) substitution on the same data split.

the language representation, and thus further improvements might facilitate language model deployment with less data engineering requirements.

As no labelled ground truth evaluation dataset exists, multiple weaker evaluation methods were used to improve the reliabil-

ity of the results. If the results indicate similar conclusions despite the different evaluation methods, it is more likely that they correlate with some underlying truth which would otherwise be represented by ground truth labels. In principle, the normalised K-Means score only conveys information on the relative average distance between cluster data points and their epicenters. Randomly distributed data can in theory result in a lower K-Means score than perfectly clustered data if K is smaller than the number of clusters, for instance with two distinct separate clusters and $K = 1$. However, coupled with a visual inspection and comparison in Figure 3, it is clear the reduction is due to improved representation rather than an under-fitted K . This is also evident from Figure 5, where the K-Means score is higher after substitution when $K = 1$ and 3, and where considerable improvement is seen first after $K = 6$. The improved performance of the LSTM-system with embedding inputs is also important for evaluation as it indicates a better distribution of the embedding space with regard to fault classes. It also reinforces the concept of assisting language model evaluation through a keyword-based system, as the substitution removes the keywords from the language model input space. Consequently, the improvements seen in the classification step must be from a superior distributional representation of the underlying semantics rather than a "short-circuit" mapping between keywords.

The lack of intrinsic or extrinsic evaluation tools or datasets for language models in the technical domain obfuscates potential improvements in technical language understanding. [Cadaid et al. \(2020\)](#) evaluated technical language representations using internal properties of the dataset, with equipment descriptions, symptoms and equipment importance as input, and type of disturbance (dominant or recessive) and maintenance workload (hours) as outputs. The existence of these two possible outputs facilitates an extrinsic evaluation, though one not necessarily in complete correlation with annotation semantics, as their results indicate that the TF-IDF-based methods by far outperform pre-trained CamemBERT and often perform just shy of fine-tuned CamemBERT. Thus, the increased natural language understanding of CamemBERT is either impeded by OOV technical terms, or the evaluation method has poor correlation with language understanding. Without access to resources such as GLUE ([Wang et al., 2018](#)) and SuperGLUE ([Wang et al., 2019](#)) from NLP, it is difficult to properly evaluate whether the language model is overfitted to the specific language distribution.

[Nandyala et al. \(2021\)](#) used an English dataset without obvious extrinsic evaluation tasks, and thus rely on quantitative evaluation in word similarity, sentence similarity and word cluster projections to compare their different distributional word vector models. These methods offer some initial insight into the workings of language models on technical language, but without quantitative values it is difficult to compare models and evaluation relies on subjective human judge-

ment. Ideally, work towards creating a technical language version of GLUE and SuperGLUE can be initiated to further the research into adaption, pre-training and fine-tuning of language models on technical language.

5. CONCLUSION

This study has investigated the effect of substituting out-of-vocabulary technical terms with natural language descriptions on BERT and SentenceBERT word distributions. To evaluate the models, a keyword-based labelling system was designed and used for visualisation and comparison with a K-Means clustering algorithm. Furthermore, the system was used to generate labels for optimization and testing of an LSTM with two output layers. The K-Means scores of clusters formed from t-SNE and PCA transformations of SentenceBERT representations of condition monitoring annotations were also computed and used as evaluation metrics. We contribute to the methodology of evaluating machine learning models' understanding of language, which was investigated by generating multiple evaluation methods where no ground truth is present.

There are many opportunities to bridge the gap between NLP successes and technical language challenges, but one major factor impeding this progress is the lack of standardised evaluation resources on technical language. For further research, we suggest the development of a technical version of the GLUE benchmark, and additional experimentation in transferring natural language understanding from large pre-trained language models to small technical language datasets through either data manipulation, fine-tuning or other transfer learning approaches. The effect of technical language substitution could also be further investigated through a systematic study of the effect of each substitution and different sets of substitutions, ideally on a dataset with many OOV technical terms but with some type of ground truth labels available for evaluation.

ACKNOWLEDGEMENTS

This work is supported by the Strategic innovation program Process industrial IT and Automation (PiIA), a joint investment of Vinnova, Formas and the Swedish Energy Agency, reference number 2019-02533. The technical term descriptions used in this work were kindly provided by Håkan Sirkka. We thank the members of the project reference group including Per-Erik Larsson, Kjell Lundberg, Håkan Sirkka and Peter Wikström, for valuable inputs.

REFERENCES

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *ICLR 2015*. (First published as pre-print on arXiv.)

- Brundage, M. P., Sexton, T., Hodkiewicz, M., Dima, A., & Lukens, S. (2021). Technical language processing: Unlocking maintenance knowledge. *Manufacturing Letters*, 27, 42–46.
- Cadavid, J. P. U., Grabot, B., Lamouri, S., Pellerin, R., & Fortin, A. (2020). Valuing free-form text data from maintenance logs through transfer learning with CamemBERT. *Enterprise Information Systems*, 0(0), 1–29.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT 2019* (pp. 4171–4186). (First published as pre-print on arXiv.)
- Gage, P. (1994). A new algorithm for data compression. *The C Users Journal archive*, 12, 23-38.
- Giorgi, J., Nitski, O., Wang, B., & Bader, G. (2021, August). DeCLUTR: Deep contrastive learning for unsupervised textual representations. In *ACL-IJCNLP 2021* (pp. 879–895). Online: Association for Computational Linguistics.
- Hinton, G. E., & Roweis, S. (2002). Stochastic neighbor embedding. *NIPS 2002*, 15.
- Hochreiter, S., & Schmidhuber, J. (1997, 12). Long short-term memory. *Neural computation*, 9, 1735-80.
- Hodkiewicz, M. R., Batsioudis, Z., Radomiljac, T., & Ho, M. T. (2017). Why autonomous assets are good for reliability—the impact of ‘operator-related component’ failures on heavy mobile equipment reliability. In *Annual conference of the PHM Society* (Vol. 9).
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *ArXiv, abs/1907.11692*. (First published as pre-print on arXiv.)
- Löwenmark, K., Taal, C., Schnabel, S., Liwicki, M., & Sandin, F. (2021). Technical language supervision for intelligent fault diagnosis in process industry. *arXiv e-prints*, arXiv:2112.07356.
- Malmsten, M., Börjesson, L., & Haffenden, C. (2020, July). Playing with Words at the National Library of Sweden – Making a Swedish BERT. *arXiv e-prints*, arXiv:2007.01658.
- Nandyala, A., Lukens, S., Rathod, S., & Agarwal. (2021, Jun). Evaluating word representations in a technical language processing pipeline. *PHM Society European Conference*. 6.
- Ottermo, M. V., Håbrekke, S., Hauge, S., & Bodsberg, L. (2021, Jun). Technical language processing for efficient classification of failure events for safety critical equipment. *PHM Society European Conference*. 6.
- Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP 2014* (pp. 1532–1543).
- Peters, M. E., Ammar, W., Bhagavatula, C., & Power, R. (2017, July). Semi-supervised sequence tagging with bidirectional language models. In *ACL 2017* (pp. 1756–1765). Vancouver, Canada: Association for Computational Linguistics.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018, June). Deep contextualized word representations. In *NAACL-HLT 2018* (pp. 2227–2237). New Orleans, Louisiana: Association for Computational Linguistics.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners.
- Reimers, N., & Gurevych, I. (2019, November). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *EMNLP-IJCNLP 2019* (pp. 3982–3992). Hong Kong, China: Association for Computational Linguistics. (First published as pre-print on arXiv.)
- Rekathati, F. (2021). The KBLab blog: Introducing a Swedish sentence transformer.
- Schroff, F., Kalenichenko, D., & Philbin, J. (2015, Jun). Facenet: A unified embedding for face recognition and clustering. *CVPR 2015*.
- Schuster, M., & Nakajima, K. (2012). Japanese and korean voice search. In *ICASSP 2012* (p. 5149-5152).
- Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In I. Guyon et al. (Eds.), *NIPS 2017* (Vol. 30). Curran Associates, Inc. (First published as pre-print on arXiv.)
- Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., ... Bowman, S. (2019). Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *NIPS 2019* (Vol. 32). Curran Associates, Inc.
- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (2018, November). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *EMNLP 2018* (pp. 353–355). Brussels, Belgium: Association for Computational Linguistics. ((First published as pre-print on arXiv.))
- Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... Dean, J. (2016). Google’s neural machine translation system: Bridging the gap between human and machine translation. *ArXiv, abs/1609.08144*.

A Design Methodology for Robust Model-Based Fault Diagnosis Schemes and its Application to an Aircraft Hydraulic Power Package

Felix Mardt¹, Phillip Bischof² and Frank Thielecke³

^{1,2,3} *Institute of Aircraft Systems Engineering – Hamburg University of Technology, Hamburg, 21129, Germany*
felix.mardt@tuhh.de
phillip.bischof@tuhh.de
frank.thielecke@tuhh.de

ABSTRACT

In a system's design phase, where knowledge about the actual behavior of the system is shallow, the design of an efficient and robust system diagnostics is a complex task. In order to simplify this process, this paper presents a model-based methodology for the design of fault diagnosis schemes. The methodology analyzes the structure of available behavioral models of the system and proposes minimal sets of sensors to fulfill diagnostic requirements. In order to choose an optimal set of sensors, the method evaluates the sets in terms of costs and diagnostic robustness. The proposed fault detection, isolation and identification schemes rely on the robust evaluation of model-based residuals using Monte-Carlo methods and highest density regions to account for measurement and parameter uncertainty. To show the design capabilities, the presented method is applied to an aircraft hydraulic power package and the resulting schemes are tested on a real test rig.

1. INTRODUCTION

The highly competitive nature of the aviation industry requires the optimization of every aspect of an aircraft's life cycle. Thus, the optimization of aircraft maintenance being one of the biggest contributors to the direct operating costs is a prominent research field. Several new concepts such as condition-based, predictive or prescriptive maintenance have been developed which aim at a condition or health oriented maintenance in contrast to the historical preventive maintenance strategies. The key enabler for all of these new strategies is on-board fault diagnosis. Fault diagnosis is the umbrella term for the full sensor-based health assessment and consists of fault detection, isolation and, if applicable, identification (FDII). The design of an efficient and robust fault diagnosis scheme, which includes the selection of sensors as

well as the algorithms for the different layers, is a difficult task. Especially during the design phase of complex systems where the knowledge about the actual behavior of the system is shallow, it is not trivial to find a set of sensors which fulfills the robustness and FDII requirements in an optimal manner.

To support this task, this paper presents a methodology for the design of a robust model-based fault diagnosis scheme which supports the designer by utilizing knowledge contained in behavioral models of the system. These models are generally available in modern model-based engineering processes and can be exploited for diagnosis design during a system's design phase. This shortens the overall development by saving design iteration and testing time.

This paper is structured as follows. Section 2 describes the chosen concept of how models are used to diagnose a system, as well as the implementation of these concepts to a system as an FDII engine. The methodology to derive an FDII engine from a behavioral model of the system is presented in Section 3. To assess the applicability, the concept is applied to a hydraulic power package in Section 4. The paper closes with conclusions and remarks in Section 5.

2. MODEL-BASED FAULT DIAGNOSIS

As mentioned in the introduction, behavioral models are a sound source of knowledge during a system's design phase, which originates from the physical base most behavioral models are built on. These physical relations are therefore explainable and comprehensible. This comes with the downside of generally complex, non-linear and dynamic equations, which require costly calculations to solve. Therefore, a lot of FDII methods rely on linear models, which are generally less computationally expensive and allow the use of the well understood theory of linear systems. Since linear models are usually only applicable in a limited space of operation and the scope of the methods discussed in this paper is maintenance FDII which is usually not as time critical as its safety related counterpart, the full non-linear equations shall be used.

Felix Mardt et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Another downside of using these non-linear dynamic models is that their solution depends on initial values and convergence. Thus, making it not only computationally expensive but limiting their robustness. It is assumed that most of the maintenance-relevant faults can be detected using steady-state relations which ignore the dynamics and remove the complexity of solving dynamic models. Thus, the chosen method shall use full non-linear steady-state models. The following section describes the utilization of non-linear steady-state equations for FDII in general and how the general method is applied to an actual system.

2.1. Utilizing Non-Linear Models for Fault Detection Isolation and Identification

Consider a model M of some physical process P

$$P \sim M = \{e, x, y, \theta, f\}, \quad (1)$$

where M consists of equations e with unknown internal states x , known measurements y , parameters θ and potential faults f . To use this model for FDII purposes, there has to be analytic redundancy. This means that there are parts of the model or subsystems of the form

$$M^* = \{e^*, x^*, y^*, \theta^*, f^*\} \quad (2)$$

such that the subsystem M^* contains more equations e^* than unknowns x^* .¹ This property is called analytic redundancy since there are more equations than needed to calculate the unknowns. If the degree of redundancy is exactly one, meaning that $\text{card}(e^*) - \text{card}(x^*) = 1$, the subsystem is called minimally over-determined and a single test can be formulated. The test used here uses input-output models which utilize the set or system of equations $\{e^* \setminus e_i\}$ to calculate all x^* and the remaining equation e_i is used to test the system for consistency. The simplest way of doing this is by subtracting the left (LHS) from the right-hand side (RHS) of e_i and forming a residual

$$r(y^*, \theta^*, x^*) = \text{RHS}(e_i(x^*, y^*, \theta^*)) - \text{LHS}(e_i(x^*, y^*, \theta^*)). \quad (3)$$

For the sake of simplicity, the term residual will also include the calculation of x^* for the rest of the paper. If the measurements y^* are consistent with the model, the residual returns zero, since the left and right-hand side are equal. If the measurements aren't consistent with the model, the residual returns a value different from zero. If a connection between faults f^* and equations e^* of the subsystem has been defined, a residual can be used to test the system for the faults affecting its equations. For the defined subsystem above, a residual value different from zero would be an indicator that one of the faults f^* might be present.

Ideally, the defined residuals are equal to zero in the fault-free

case and different from zero in the case of a fault. In reality, however, this is not the case. Due to modeling errors and uncertainty in the measurements as well as parameters, the residuals are in practice almost always different from zero. To solve this problem, we published a method in (Mardt & Thielecke, 2021) which allows for a physical based statistical evaluation of the residuals under uncertainty to facilitate robust fault detection. The method uses uncertainty estimations for the parameter and sensor uncertainty gained from a-priori knowledge as well as multiple sensor readings. These uncertainty distributions are used to sample a set of possible residual values using Monte Carlo simulation (MCS). Thus, rather than single values for r , samples of the set of possible values S_r are generated. The test for consistency then becomes

$$\begin{aligned} 0 \in S_r & \text{ if } y^* \text{ consistent with } M^* \\ 0 \notin S_r & \text{ otherwise} \end{aligned} \quad (4)$$

Since S_r is not known explicitly, the samples are examined to determine whether 0 is part of the set. This is done using Highest Density Regions (HDR), a method which returns the set $S_{r,\alpha}$ which contain the most probable $100(1-\alpha)\%$ of values as interval boundaries (Hyndman, 1996). The proposed consistency test is

$$\begin{aligned} y^* \text{ consistent with } M^* & \text{ if } 0 \in S_{r,\alpha} \\ y^* \text{ inconsistent with } M^* & \text{ otherwise} \end{aligned} \quad (5)$$

E.g. for $\alpha = 0.05$ the fact $0 \in S_{r,\alpha}$ means that the measurements are consistent with the most probable 95% of all possible systems. This method ties a probability parameter to the decision which allows controlling the false alarm rate, since for this example 5% of the possible systems would lead to a detection and thus an alarm even though they are not faulty. Since this is a Monte Carlo based method which uses HDR to interpret the results, it will be subsequently called MCS-HDR.

Combining the information about which residual is sensitive to which fault, a detection and an isolation of the considered system can be conducted in a robust way using the MCS-HDR method, given that there are enough residuals. Fault identification goes further in the sense that it computes a value or range for a considered fault f to base immediate or future actions on that information. In (Mardt & Thielecke, 2021) we proposed a method which relies on the evaluation of multiple copies of the same residual with explicit fault inputs to decide which one fits best. This requires explicit fault models rather than just the information that a fault affects an equation. This method of fault identification is computationally expensive since it requires multiple MCS and HDR evaluations for the same fault. A simpler method which wasn't discussed in previous publications is the direct calculation of the fault input f using MCS HDR. When the equation e_i in Equation 3 is the one affected by a fault and is analytically solvable for that fault rather than just solving for 0, the MCS HDR method can

¹The asterisk denotes that the given sets are subsets of the full model.

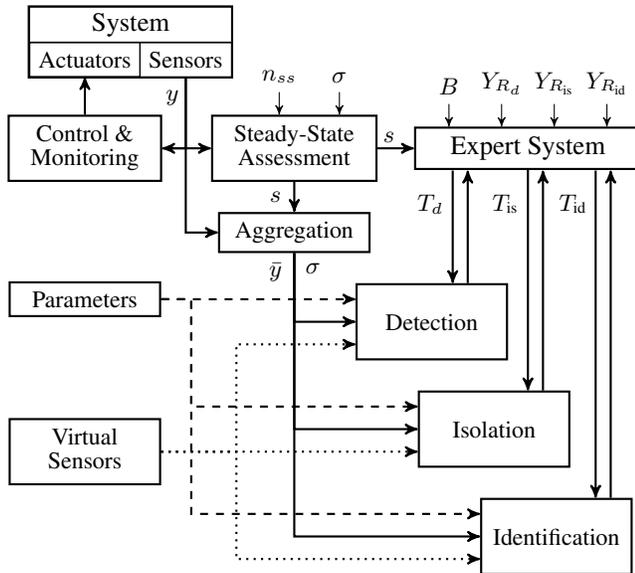


Figure 1. Proposed implementation architecture

be used to calculate the most probable intervals for the fault input directly. This new approach requires less computational power and leads to a better interpretability of the results compared to the comparison of multiple residuals with different fault inputs. Consequently, this method will be prioritized as long as the requirements are met.

2.2. Implementation of Model-Based FDI

To efficiently implement the abstract ideas of using non-linear models for FDI purposes, an actual architecture is needed. The proposed architecture is shown in Figure 1. The monitored system is shown in the top left. It consists of sensors and actuators which are connected through the control and monitoring of the system. The control and monitoring closes the control loop and ensures the safe operation of the system. All other shown blocks are part of the maintenance related FDI and will be discussed in the following paragraphs.

The first block which directly interacts with the system is the steady-state assessment block. It continuously monitors the sensor signals and determines whether a signal can be considered to be in steady-state. Since the residuals used for FDI are based on steady-state equations, they need steady-state sensor signals in order to provide valid results. The method used for steady-state detection is based on a sliding window standard deviation. Therefore, the standard deviation of each signal is constantly calculated over the last $n_{ss,i}$ values and compared to a constant. This constant is based on the expected distribution of the sampling standard deviation of a normally distributed sensor signal with known standard deviation σ_i for $n_{ss,i}$ samples. The output of the block is one binary signal s_i for each assessed measurement, which states if that signal is considered steady-state.

As soon as a signal is considered steady-state, the steady-state assessment block triggers the aggregation block. This block calculates the running mean \bar{y}_i and sample standard deviation σ_i of that signal as long as it is in steady-state and resets once it leaves steady-state. These two values per measurement are used as input to the MCS-HDR method used in the detection, isolation and identification blocks.

The expert system block is the supervising element for the FDI process. It continually receives the information about which measurements are in steady-state and holds the static information about which signals are used in which detection-residuals $Y_{R_{d,j}}$. Detection-residuals are a subset of all residuals which reliably detect all faults of the system. They are the first ones to be evaluated and set the starting point for the other diagnosis stages. When all signals were in steady-state and one of them switches state, the expert system triggers that specific detection-residual using the trigger signal

$$T_{d,j}(n) = \begin{cases} 1 & \text{if } \forall y_i \in Y_{R_{d,j}} : s_i(n-1) = 1 \text{ and} \\ & \exists y_i \in Y_{R_{d,j}} : s_i(n) = 0 \\ 0 & \text{else.} \end{cases} \quad (6)$$

The reason for that is, that this is the point in time when the most information about all the measurements has been gathered. An evaluation of the residual before that would lead to worse results.

The detection block uses the MCS HDR method explained in the previous chapter. It conducts an MCS to receive residual samples, which are then evaluated using HDR and a given α_i for each residual. Whether 0 is part of the calculated HDR is fed back to the expert system by one boolean signal for each implemented residual. In addition to the measurements sample mean and standard deviation, the detection block also receive parameter samples from a database and samples from virtual sensors. Virtual sensors are variables of the system which aren't measured but can be estimated with some uncertainty. For example, the temperature of a fluid might be needed to calculate the density of said fluid. In practice, however, the benefit of actually measuring the temperature might be slim, and it could suffice to just assume that the temperature is somewhere in a plausible range. This is what virtual sensors do, they supply samples of sensors which aren't measured but estimated and allow including the uncertainty tied to that virtual measurement.

The expert system receives the results of the MCS-HDR evaluation of the detection-residuals and enables the isolation-residuals which are needed for the next diagnosis step. This is done based on the residual fault matrix $B \in \mathbb{B}^{n_r \times n_f}$ where n_r and n_f are the number of total residuals and faults respectively. This matrix encodes the fact that a residual is sensitive to a specific fault. If the observed residual pattern from the detection-residuals suggests multiple possible single faults explaining that pattern, the respective isolation-

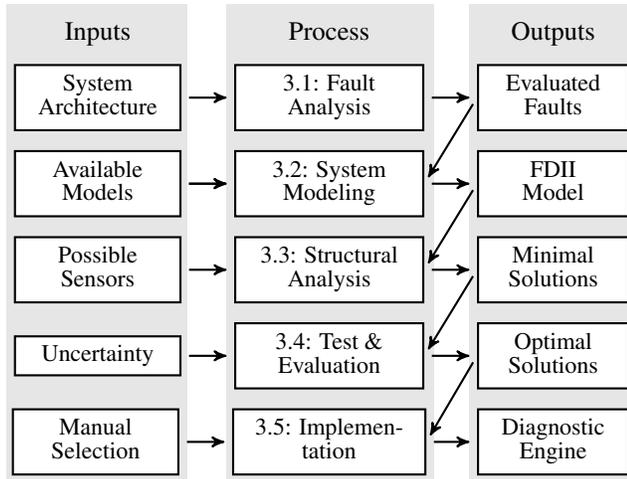


Figure 2. Design methodology

residuals are triggered. The isolation-residuals are another subset of all residuals and are needed to fully isolate all relevant faults from each other. The reason these aren't always evaluated is to save computation time, power and thus costs.

For the last step of the FDII the expert system enables the identification-residuals which correspond to the isolated faults to identify them and compute an actual health status. This is done by the identification methods discussed in the previous section. All residuals which fulfill the requirements for the direct computation of f are implemented accordingly. All other residuals use the indirect method presented in (Mardt & Thielecke, 2021).

3. A DESIGN METHODOLOGY FOR FAULT DIAGNOSIS SCHEMES

As explained in the previous section, the implementation uses certain measurements of the system to evaluate different sets of residuals for FDII purposes. The selection of these sensors and matching residuals is a difficult task, especially when the system under consideration is a complex one. To aid the system's designer in the process of choosing sensors and matching residuals during the design stage of the system, this section presents a design methodology. The general steps of this methodology are depicted in Figure 2 and chronologically examined in the following sections.

3.1. Fault Analysis

The process of designing an FDII system, similar to almost all engineering processes, starts with the definition of requirements. In this case this is a list of component faults which shall be detected, isolated and identified. The general analysis proposed here is closely related to first steps of the MSG-3 aircraft industry standard for the planning of maintenance tasks (Air Transport Association, 2002).

The analysis starts by defining maintenance significant items (MSI). These are components from which a fault leads to: an interruption of service, a decrease of operational reliability, or comes with a significant economic impact. For these MSI, a failure mode and effect analysis (FMEA) is carried out. This means evaluating which failure modes a component may show and how they impact the rest of the system's operation. The third step takes these results and categorizes the MSI failure modes based on the impact (safety, operational, economic) and the type of manifestation for the operating crew (evident, hidden).

After assessing the effects of all faults and the baseline of onboard detection, the need for additional diagnostic steps by a dedicated FDII system can be evaluated. For the detection, this includes evaluating whether the detection of a hidden fault by a dedicated monitor is potentially less expensive than the operational costs induced by the same fault. This evaluation, like all economic considerations in this context, are not the focus of this work and thus not further discussed. There is a plethora of research on this topic for the interested reader. It's important to state that at this point the actual costs of isolating or detecting a fault are not yet evaluated and will be a result of step 4 of the methodology. Thus, the economic analysis has to use rough assumptions for the costs, which can be validated later in the process. If the costs are substantially different from the assumption, an iteration loop might be useful.

The formulation of isolation requirements should be straightforward and be based on maintenance routines. If two faults lead to the same maintenance action, i.e. replacement of a specific line-replaceable unit (LRU) there is no need to isolate them on-board. If on the other hand the two faults require different maintenance actions, the isolation is necessary to avoid unnecessary actions during the maintenance process. This trade-off could also be economically studied by actually calculating the cost of a maintenance iteration and the potential cost of the on-board isolation.

Similar to the steps above, the decision of whether a fault should be identified should be based on a cost benefit analysis. For components which degrade randomly or show complex degradation patterns while having a large operational impact, the identification or health monitoring becomes more valuable.

The final result of the fault analysis step is a table containing all MSI faults which shall be detected. For each fault, an isolation group is defined which states from which other faults it shall be isolated and a binary indication on whether the fault shall be diagnosed or not.

3.2. System Modeling

The second step of the proposed methodology is the system modeling. This step sets up the FDII model, which is subsequently analyzed in the following steps. As defined in Equation 1 a model consists of equations, variables, measurements, faults and parameters. Ideally, the equations of the physical behavior are already defined in available models from the system’s design phase. In most cases, these available models only contain nominal behavior and the equations need to be extended with faulty behavior based on the fault inputs f if the requirements from the previous step include fault identification.

To simplify the modeling process and make the FDII models more maintainable, this method uses a custom JSON-based modeling environment. This object-oriented environment allows for the definition of general components which are stored in a library. These components can be instantiated and connected to build the actual system. By implementing an acausal modeling approach similar to Modelica or Simscape, the direction of physical ports is omitted and not of the users’ concern. The connecting equations based on Kirchoff’s laws are added automatically based on a defined topology of the system.

Since this step’s goal is to set up the model for the structural analysis, only the structure of the model has to be defined. The actual parameters are not yet needed.

3.3. Structural Analysis

The third step of the presented design methodology is the structural analysis of the behavioral model created in the previous step. The goal of this step is to determine which possible minimal sensor combinations fulfill the FDII requirements specified during the fault analysis using the model created in the previous step.

The use of structural analysis to determine which faults are structurally detectable and isolable goes back to the work of (Cassar & Staroswiecki, 1997). They propose to analyze a bipartite graph $G = (X, E, A)$ representing the structure of a system by mapping the variables X to the equations E they appear in by the edges A . Applying the Dulmage-Mendelsohn decomposition (DM) to G assigns each of the nodes in X and E to one of the three groups: under-determined, just-determined and over-determined. Equations in the over-determined part of E are considered monitorable. This means that they can be tested for consistency because there are more equations than variables in this part of the system, hence the name over-determined. Thus, it is structurally possible to build residuals out of these equations. The adjective structurally is important in this context because the structural analysis as it is defined for now does not take into account the actual equations of the system. Thus, solving

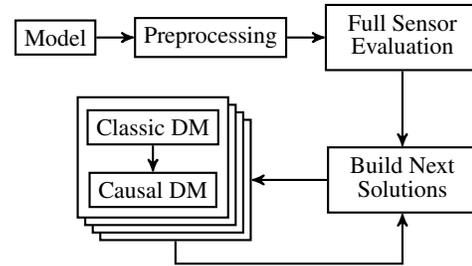


Figure 3. Sensor placement flow chart

over-determined subsystems may require numerical solvers or, in the dynamic case, integration.

The work of (Rosich, Frisk, Aslund, Sarrate, & Nejjari, 2011) takes the causal solvability of equations into account and presents a method to compute a DM which ensures causal solvability. This means the variables in the just and over-determined parts are explicitly calculable without the use of numerical solvers. To take the solvability into account, additionally to the equation itself, the information about for which variable the equation is solvable needs to be given.

The publication (Rosich et al., 2011) also adapts the sensor placement via structural analysis presented in (Krysander & Frisk, 2008) to only produce causal solvable results. Since the causal solvability poses additional restrictions on the sensor placement problem, the presented solution is a greedy brute-force search where costs of sensor sets are considered when choosing the next set to test. Thus, the placement can stop after a set has been found which fulfills the requirements, since it is the cheapest one to do so.

The sensor placement proposed here uses most of the concepts from (Rosich et al., 2011) to produce causal solvable solutions. For that reason, the library approach presented in Section 3.2 includes the information about which equations are solvable for which variables. Since the aim of this work is to also evaluate the robustness of sensor sets and thus tackle a multicriterial optimization, the causal sensor placement cannot be used as is and needs to be adapted. Due to limited space, a high level overview of the process is depicted in Figure 3 and will be discussed briefly in the following paragraphs.

To prepare the system of equations for the structural analysis, the first step of the sensor placement process is the preprocessing. This includes the elimination of redundant variables as well as mandatory transformations of the system of equations to ensure proper results. The elimination of redundant variables not only clarifies the equations and saves memory, it also reduces the number of sensor combinations needed to be evaluated by exploring equality between these sensors. If two sensors are structurally equal, only one of these sensors needs to be tested and the number of possible solutions is essentially halved. The transformation of the system of equa-

tion might be mandatory, since the structural analysis only assesses the structure of the system of equations. Consequently, the results depend on the formulation of the equations and can be altered through analytical transformation of the equation. Consider for example the two equations

$$\begin{aligned} x_1 &= x_2 + x_3 \\ x_4 &= x_2 + x_3. \end{aligned} \quad (7)$$

This system of equation contains four unknown variables and two equations, and thus structural analysis comes to the conclusion that measuring two of these variables leads to the calculation of the other two. This holds for most cases but does not when measuring x_1 and x_4 since they're equal. To find these conflicts, which often occur in flow networks, algorithms are implemented to solve the equations and transform them into structurally unique representations like

$$\begin{aligned} x_1 &= x_2 + x_3 \\ x_1 &= x_4. \end{aligned} \quad (8)$$

To assess whether the FDII requirements can be met with the available sensors and if all sensors are actually needed, the next step evaluates the full sensor model. This step is adapted from (Rosich et al., 2011) and modifies the requirements and list of possible sensors which are used in the next step.

After the preprocessing and evaluation of the model containing all sensors, the next step is the actual sensor placement. Since the goal is to find all minimal causal solvable solutions, all sensor combinations have to be evaluated, which leads to an exponentially complex problem. To account for this, the implemented algorithms are optimized for efficiency. This begins with the chosen search strategy. A breadth-first search is used, which analyzes all sensor sets of a given length first before evaluating the next length. This allows for the parallelization of the actual evaluation of the combinations, since the results of each evaluation do not affect the ones on the same level. Evaluating multiple solutions in parallel is one key element of making the evaluation of hundreds of thousands combinations possible. The other key element is the implemented evaluation strategy. Instead of evaluating the causal DM of the currently evaluated structure directly, a two-step method is applied. This tests the classic DM without the causality restrictions first and only if that fulfills the requirements, the much more computationally expensive causal DM analysis is conducted. Since a large amount of tested solutions does not fulfill the requirements, this approach immensely increases the efficiency. If a valid solution is found, this solution will not be expanded in the next step, since it is by definition a minimal one. The generation and evaluation of new solutions is repeated until no new solutions are possible.

3.4. Evaluation

After calculating all possible sets of sensors, they can be evaluated to choose an optimal one for the task at hand. The first and most straight forward is the evaluation of costs. This step sums the costs of each sensor in each combination and returns a single value for each set. The cost of each sensor is considered an input to this method and depends not only on the initial cost of the sensor itself and its installation, but also on the running cost of evaluating and maintaining that sensor.

The next evaluation step is the evaluation of the FDII performance of each sensor set. For each of the detection, isolation and identification steps, measures can be defined to assess the performance. For the detection, these are the rates of a false detection, namely the false alarm rate (FAR) and the missed detection rate (MDR). As explained in Section 2.1 the FAR considering a random system inside the defined bounds is defined by the α value used in the HDR evaluation of the residuals. Consequently, this rate is an input parameter rather than something to evaluate. The MDR on the other hand is not known until evaluated in some form. To test the MDR, some faulty data is needed to use as input into the implemented residuals and assess if they detect the fault. In the development process of the system real data is generally not available and even if a prototype has been built it rarely produces the faulty data needed to test the residuals. The easiest way to get faulty data is the available model from the design process, which has been used to derive the diagnosis model. It's also possible to get data from the diagnosis model, but since it wasn't built for simulation and is not necessarily entirely solvable, this is a more tedious task.

To generate the faulty data, faults have to be injected into the system. It is assumed that all faults are easier to detect the higher their fault input f_i is. Thus, to evaluate the performance a minimum detection requirement for each fault has to be specified which is the fault value at which the data is created and the MDR evaluated.

Given that faulty data of the system is available, the residuals can be implemented and tested. To implement residuals, the over-determined part of the system including the set of sensors needs to be examined. This is done by calculating causally minimal structurally overdetermined sets of equations (MSO). These are the minimally overdetermined subsystems discussed in Section 2.1. There are usually plethora of MSO for each solution, and they all detect a subset of faults and can be implemented as a residual. To evaluate the MDR for a fault, all possible residuals that detect this fault have to be tested. The lowest MDR of all residuals for each fault is the best possible MDR for that solution. Note that it is assumed here that the number of implemented residuals is of no concern, and thus the best MDR can be achieved even if one residual for each fault has to be used. The detection evaluation results in the best achievable MDR for each fault for

each solution.

Note that the MDR depends on the chosen FAR, in a way that a higher FAR will lead to a lower or equal MDR. The correlation between the two depends on the width of the uncertainties. If the fault leads to a behavior which is entirely separated from the nominal behavior, both an FAR of 0 and an MDR of 0 are possible. If the faulty and nominal behavior overlap, this is not possible and a trade-off between the two has to be chosen.

To assess the isolation performance, the same MDR data is used as for the detection performance. Rather than assessing the MDR of all residuals for one fault, the MDR of each isolation combination is considered. I.e. if f_1 and f_2 are supposed to be isolated all residuals which detect f_1 but not f_2 are considered and the lowest MDR is taken for this category. Thus, the isolation evaluation results in one best achievable MDR for each required isolation combination for each possible solution.

The evaluation of the identification performance can also be conducted by applying the residuals to faulty test data. In this case, there are two evaluation criteria: the rate of correct classification and the accuracy in terms of an average width of the predicted health interval. The second criterion is only reasonable for the direct implementation of the fault identification, where an actual interval is calculated. For the indirect implementation, which assesses multiple fault intervals for the same residual, the width of the tested intervals is an implementation parameter.

3.5. Implementation

The Implementation is the last step of the proposed design methodology. It takes the chosen solution(s) and defined parameters and automatically generates the FDII engine seen in Figure 1. The engine is implemented into a Simulink model which can be connected to the design model for further testing or used for code generation to apply to a test rig for online execution.

4. APPLICATION

The following section describes the results of the application of the previously presented design methodology to a hydraulic power package (HPP). The section begins by introducing the HPP with all its components, purposes and operation modes. The following parts are structured according to the methodology, beginning with the fault analysis which leads to the modeling of the HPP followed by the structural analysis. The results from the analysis are subsequently analyzed and promising solutions are chosen for the final part of this section, the implementation and application to data from the actual HPP test rig.

4.1. System description

The HPP is a compact unit integrating two redundant pumps and the necessary hydraulic system equipment (reservoir, filters, valves). Due to the compact and modular design, it can be integrated into modern, More Electric Aircraft architectures to supply local hydraulic circuits. The modular approach enables operational benefits for installation, maintenance and testing. (Trochermann, Rave, Thielecke, & Metzler, 2017)

The system architecture of the HPP is depicted in Figure 4. The two redundant motors, powered and controlled via an external power electronic unit, supply mechanical power to their connected pumps. The pump turns the mechanical into hydraulic power and pushes hydraulic fluid through a check-valve, which prevents flow reversal during one pump operation. After the check valves, the flows of the two pumps combine into a single one which flows through another check valve and the high pressure filter. After the high pressure filter, the fluid flows to the consumers via the high pressure port (HP). In case of a malfunction, the fluid can bypass the consumers through the pressure relief valve (PRV). The return flow arrives at the low pressure port (LP) and flows through the low pressure filter before entering the reservoir. From the reservoir, the fluid is sucked through a flow restriction. Every one of the components shown in Figure 4 is connected to the others via pipes not shown in the schematic.

For clarity, only measurement points in contrast to the actual sensors are depicted in Figure 4. The measured quantities for each of the points are listed in Table 1, where T is the temperature, p is the pressure, dp is the differential pressure, u is the voltage, i is the current, ω is the angular velocity and \dot{V} is the volumetric flow rate. The shown sensors are the ones installed on the test rig at the Hamburg University of Technology. Most of these sensors are installed for testing purposes and not necessarily needed for the operation of the HPP. The minimum set of sensors needed for a nominal operation depends on the chosen control and operating strategy. One possible set taken from (Trochermann & Thielecke, 2021) is subsequently used and includes the following sensors $\{T_{s2}, p_{s2}, u_{s7,8}, i_{s7,8}, \omega_{s9,10}, p_{s11,12}\}$.

The HPP comprises redundant motor pumps for reliability purposes. This means that the HPP is sized to supply the connected hydraulic system with only one operable pump. Thus, in almost all modes of operation, only one pump is actively driving the connected system while the other one is on standby. Only in some, not safety-critical modes, both pumps are used to increase the delivered flow of the HPP. Consequently, only one pump operation is considered in the following analysis, since it massively reduces the system complexity and thus sensors and algorithms needed for the FDII of the HPP.

Note that the HPP test rig was not built to design or test FDII

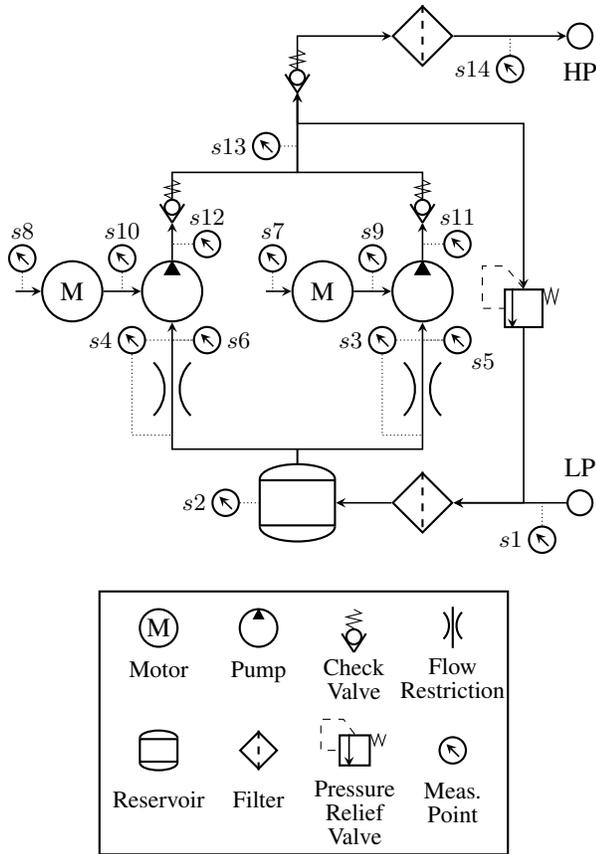


Figure 4. Hydraulic power package system schematic

methods and thus does not offer the possibilities to emulate faults of the system. The installed sensors were not chosen to diagnose the HPP, but rather to design control strategies and to validate models of the system.

4.2. Fault Analysis

As described in Section 3 the first step of the presented method is the analysis of the faults of the system and the respective need for detection, isolation and identification. Basically, all the depicted components in Figure 4 are subject to degradation and need to be replaced at some point of the HPP life. It is important to keep in mind that a continuous observation of a component produces costs in the form of initial invest-

Table 1. Measurement points and quantities of the HPP

Sensors	Measured Quantities
s1, s5, s6	T
s2, s14	p, T
s3, s4	dp
s7, s8	u, i
s9, s10	ω
s11, s12	p
s13	\dot{V}, p, T

Table 2. Results of the HPP fault analysis

Fault Name	Isol. Group	Identified
$f_{hm,pump}$	1	yes
$f_{vol,pump}$	2	yes
f_{motor}	3	yes
$f_{lp,filter}$	4	no
$f_{hp,filter}$	5	no

ments and running costs for the onboard sensors and computing power. Thus, for components which degrade on a large time frame, a continuous FDII produces more costs than it saves in maintenance. As explained above, the economic analysis of this is not part of the methodology of this paper. Consequently, the components and their FDII level have been selected according to engineering judgment.

The faults considered relevant for FDII purposes are listed in Table 2. The first one is the hydro-mechanical degradation of the pump $f_{hm,pump}$. This fault affects the ability of the pump to turn mechanical power on the shaft to differential pressure across the pump. It describes the torque loss due to friction on the mechanical and hydraulic side, hence the name. The second relevant fault is the volumetric degradation of the pump named $f_{vol,pump}$. This fault decreases the ratio between theoretical flow of the pump given by its speed and the actual hydraulic flow in the system. It is a measure for the amount of internal leakage in the pump. The third fault is a motor fault named f_{motor} . It is a lumped fault which affects the efficiency of the motor to turn electrical into mechanical power delivered to the pump. It combines mechanical friction as well as electrical losses in the motor. The fourth and fifth fault $f_{lp,filter}$ and $f_{hp,filter}$ represent the clogging of the filter on the low and high pressure side respectively. All of these faults shall be detected and isolated from each other. The reason for the isolation definition is to separate different faulty LRU to facilitate maintenance. The identification is limited to the motor and pump, since the degradation of these components is much more complex than the filter degradation and might benefit from a continuous health monitoring.

4.3. System Modeling

As described in Section 3.2, ideally, there is already a model of the system under consideration present when starting the process of FDII design. This is the case for the HPP as well. Two models, one for the controller design and one for the thermal analysis of the HPP, are available. Both of these models model nominal dynamic behavior of the HPP and have been validated for their specific purpose using the real test rig. Having an already validated model is an ideal starting point to build an FDII system for that specific unit, but does only provide a small benefit when it comes to a general FDII scheme for all units of that kind.

Since both available models were built for different purposes,

their modeling scope is different. Therefore, the most accurate part of both models is combined to build a single steady-state model for FDII design. The resulting model uses the library concept described in Section 3.2 to model three different domains: an electrical one to represent the electrical interface of the HPP, a mechanical one to model the connection between the motor and the pump, as well as a hydraulic domain with temperature dependent fluid parameters to model the rest of the system. For most of the components, the equations used in one (or both) of the two existing models were used. For other components, which were out of scope in both of the models (i.e. the filters), manufacturer’s information was used to build new models. The flow restriction is the only component which uses a polynomial fitted to test rig data to model the pressure loss, since it does not behave like a standard hydraulic component, i.e. an orifice.

None of the available models depicts faulty behavior of the components, which is why this had to be added to the equations. In contrast to the physical modeling of the components, the fault modeling approach is a more practical one. The hydro-mechanical pump and motor faults use an efficiency based approach

$$P_{out} = (\eta_{nom} - f)P_{in} \quad (9)$$

where P_{in} and P_{out} are the in and output power respectively, η_{nom} is the nominal efficiency and f is the fault input affecting the behavior. Thus, identifying f is directly related to the efficiency, an easily interpretable quantity. A similar approach is used for the volumetric degradation of the pump

$$\dot{V}_{act} = (\eta_{nom} - f_{vol,pump})\dot{V}_{th} \quad (10)$$

with a nominal volumetric efficiency η_{nom} and the actual and theoretical flow \dot{V}_{act} and \dot{V}_{th} respectively.

4.4. Structural Analysis

To analyze the structure of the model of the HPP and find applicable sets of sensors to fulfill the FDII requirements, the possible sensors have to be defined. The aim of this work is to test the resulting sensors sets on the real test rig. This comes with the restriction, that only sensors can be used which are available on the real test rig. Therefore, the list of possible sensors is the list of sensors installed on the real test rig listed in Table 1. To compute the sets of possible sensors, the approach presented in Section 3.3 is used, and the results are discussed in the following section.

The first step of the sensor placement process is the preprocessing. During this process, the model is simplified and redundant variables and sensors are discovered. For the HPP, this step reveals that the sensors T_{s13} and T_{s14} measure the same quantity. This comes from the fact that no heat loss is modeled in the piping, hp filter and check valve between s_{13} and s_{14} . This means only one of these sensors has to be con-

sidered during the sensor placement, and the results can be expanded for the other sensor.

The evaluation of the model containing all sensors is the second step of the sensor placement and shows that a degradation of the filter on the low pressure side of the HPP is not possible. This is due to the fact that there is no pressure sensor upstream of the filter, which does not allow calculating the differential pressure by any means. Thus, the actual differential pressure cannot be compared to the theoretical and consequently this fault cannot be detected. All other faults are structurally detectable. The isolability analysis shows that even with all possible sensors in place, a degradation of the motor f_{motor} cannot be isolated from a hydro-mechanical fault of the pump $f_{hm,pump}$. The reason for that is that both faults act on the power conversion from electric to mechanical and then hydraulic. Since there is no full measurement of the mechanical power in between the units, the faults cannot be separated. It would need an additional torque sensor at $s_{9,10}$ to differentiate between the faults. This fact is acceptable since the motor and pump are part of the same LRU, so replacing one inevitably means replacing the other as well. This also implies, that a fault identification of both faults is not possible. One solution for this would be to combine both faults into a single one on the modeling side.

The full sensor analysis also identifies sensors which are not needed for the achievable FDII requirement. Since $f_{lp,filter}$ is not detectable, the sensor T_{s1} is not needed to fulfill the updated FDII requirements and will be removed for the sensor placement. The reason this sensor is not identical to T_{s2} is that the fluid temperature in the reservoir is considered different from the one in the return line, which makes T_{s1} obsolete.

The structural sensor placement returns 30 possible sets of sensors, which are listed in Table 3. The actual number of possible sets has to be increased to 54 since 24 of the 30 solutions contain the sensor T_{s13} which can be replaced with T_{s14} as explained above. The following analysis omits the additional 24 solutions since they are structurally and in terms of uncertainty identical to their counterparts. Due to the limited space available here, an explicit in-depth verification of all possible sensor sets will not be provided. Instead, the following paragraphs will discuss why some sensor combinations are present in the solution sets.

To detect a volumetric pump fault, the theoretical and actual flow rate need to be compared to test the Equation 10 for consistency. The theoretical flow rate \dot{V}_{th} can be computed using the pump’s displacement and the angular velocity measuring ω_{s9} . If the angular velocity is not directly measured, it can be computed using the motor equations and measuring the voltage u_{s7} and current i_{s7} . The actual flow rate can be measured directly by using \dot{V}_{s13} . An alternative way is to compute it using the measurements dp_{s3} and T_{s5} as well as the flow restriction equations to get the flow rate. This low pressure flow

Table 3. Sensor overview

s	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
T_{s2}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	
T_{s5}	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	0	0	1	1	0	1	1	1	1	
T_{s13}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	1	0	1	1	1	
\dot{V}_{s13}	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
dp_{s3}	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
i_{s7}	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	1	0	1	1	
ω_{s9}	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	0	1	1	1	1	0	1
p_{s11}	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	
p_{s13}	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
p_{s2}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
p_{s14}	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
u_{s7}	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	1	1	0	0	1	1	0	

rate needs to be converted to the pump's high pressure flow rate by calculating the mass flow through the system by measuring the reservoir temperature T_{s2} and turning that into the high pressure flow rate by measuring the high pressure temperature T_{s13} . Another option is to measure the flow's impact instead of calculating or measuring it explicitly. Thus, by measuring two of the high pressure sensors $p_{s11,13,14}$ the theoretical pressure difference between them due to the theoretical flow can be compared to the actual pressure difference measured. Thus, the sensor sets needed for the detection of the volumetric pump fault are the ones fulfilling the expression

$$\begin{aligned}
 & (\omega_{s9} \vee u_{s7} \wedge i_{s7}) \wedge \\
 & (\dot{V}_{s13} \vee dp_{s3} \wedge T_{s5} \wedge T_{s2} \wedge T_{s13} \vee \\
 & (p_{s11} \wedge p_{s13} \vee p_{s11} \wedge p_{s14} \vee p_{s13} \wedge p_{s14})). \quad (11)
 \end{aligned}$$

In fact, all listed solutions are valid with respect to this expression. Similar expressions can be set up for the detection and isolation of the other components. This considerably small example shows that this is a tedious and error-prone task to do manually, which is why the structural analysis is a valuable tool.

4.5. Evaluation

The first step of the evaluation is to assess the costs of all sensor combinations. Therefore, costs for each sensor have to be defined. For the sake of simplicity, only the procurement costs are taken into account and costs for maintaining the sensors are neglected. The costs for sensors which are considered already installed in the system are set to 0. The normalized resulting costs of all sensor sets can be seen on the sensor axis of Figure 5. These costs are widely spread, ranging from 0.36 (solution 1) to the maximum cost (solutions 28,29 and 30).

To choose a sensor set, the presented methodology not only uses the costs of the sensor sets, but also their robustness. For the detection problem, this is measured using the minimum achievable MDR of each sensor set. To test the sensor sets, a steady-state model of the whole system is used to generate

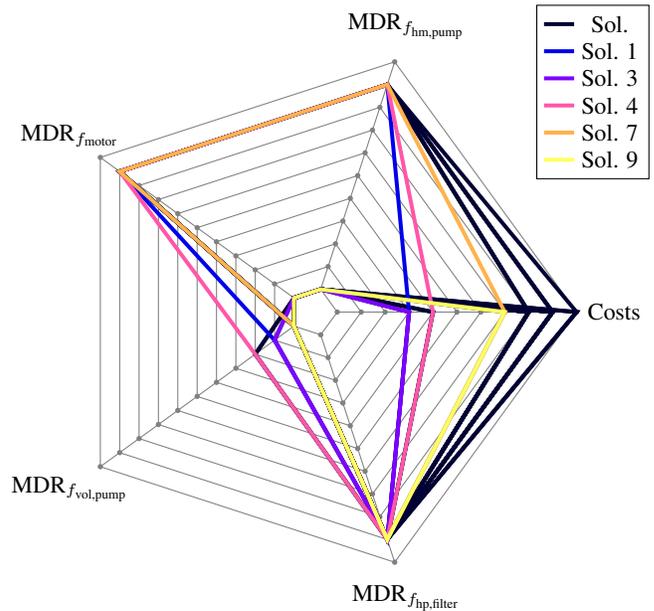


Figure 5. MDR and normalized costs of all 30 solutions with axis ranging from zero to one from the inside out.

faulty test data. The parameter and sensor uncertainties of the test model are the same as the model uncertainties and the minimum detection requirement is set to 0.2 for all faults but the volumetric pump fault, which is set to 0.3. The resulting MDR are shown on their specific axes in Figure 5. Solutions that will be discussed in the following are colored.

It is apparent that all the solutions have a high MDR for the high pressure filter faults. This is due to the relatively low accuracy of the pressure sensors p_{s11} , p_{s13} and p_{s14} combined with the uncertainties of the other flow restrictions (check-valves and pipes) in between the sensors. The deviation in pressure drop due to the required minimal detectable fault is lower than the combined uncertainty of the sensors and parameters, and thus the defined detection specification cannot be met. For this reason, most applications of filter clogging detection use differential pressure sensors, which are much

more accurate than the difference of two absolute pressure sensors. This option wasn't considered here, since such a sensor is not installed on the test rig.

Particularly high MDR rates can also be observed for some solutions for the hydro-mechanical pump and motor faults. A comparison of the sensors used by these solutions shows that all the poorly performing solutions use a combination of u_{s7} and ω_{s9} to calculate the theoretical differential pressure of the pump to compare it with the actual differential pressure applied to the system. Since the torque of a motor and thus the differential pressure of the connected pump is mainly driven by the current through the motor, measuring i_{s7} directly results in less uncertainty than calculating i_{s7} from u_{s7} and ω_{s9} . In fact, the uncertainties in the parameters involved in the calculation are too high to make these solutions feasible. An analysis of these parameters shows that the magnetic flux linkage parameter of the motors is the reason for these uncertainties. This parameter is modelled as a uniform distribution spreading 10% around the most probable value. Decreasing this uncertainty alone can turn the originally unfeasible into a feasible one.

The achievable MDR for the volumetric pump faults range from around 0 to 20% forming one cluster each at 0, 10 and 20%. The cluster at 0% contains all sensor sets which measure the volumetric flow \dot{V}_{s13} directly. This introduces less uncertainty into the detection than calculating this flow from differential pressure measurements or measuring the effect of the flow on the high pressure side, and thus shows the least MDR. The second cluster around 10% contains the solutions which measure the effect of the actual flow in the system by using the pressure differential of the sensors p_{s11} and p_{s14} (solutions 1,2,3). These solutions perform better than the ones using p_{s13} and p_{s14} due to the fact that more components between the sensors lead to a higher overall pressure drop and thus a more accurate detection. The sensor sets using p_{s13} and p_{s14} (solutions 4,5,6) form the cluster around 20% and show the worst performance. Both, the residuals using the measured pressure differential on the high pressure side and the ones using the differential pressure dp_{s3} to compute the actual flow show a similar performance.

Due to the high relative price of a volumetric flow sensor, the best performing solutions for the volumetric flow fault are also the most expensive ones. Thus, it might be beneficial to further analyze the other solutions when lowering the requirements. This is done for three of the solutions (1,4,7) - each from one of the discussed clusters - in the next paragraphs.

The MDR calculated above depend on the minimum detection requirements defined for each fault. To assess how the MDR changes and if some solutions with higher MDR become feasible when the detection requirement is lowered, an analysis is carried out. The results of this analysis are shown in Figure 6. It shows the expected behavior that when the

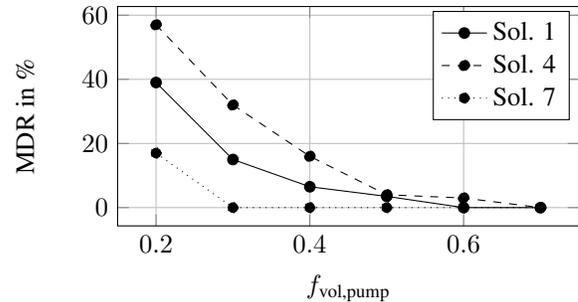


Figure 6. Missed detection rates for the volumetric pump fault for the solutions 1, 4 and 7 based on the minimum detection requirement for a $\alpha = 0.05$

requirement is increased, the MDR decreases for each of the solutions. It can be seen that lowering the requirement would increase the MDR of solution 7 to an infeasible value. Increasing the requirement on the other hand leads to substantial lower MDR for the other solutions, which could make them feasible if the higher detection requirement is acceptable. This shows that the minimum detection requirement has a severe impact on the MDR and has to be assessed carefully when choosing or dismissing solutions.

As stated in Section 3.4, the MDR depend on the selected α for the HDR calculation. For the rates shown above the selected α is 0.05 resulting in an FAR of 5%. Analyzing the effect of α on the resulting MDR of a solution can help to optimize the solutions and assess their potential. Figure 7 shows the impact of the FAR on the MDR. The general expected behavior - a decrease in the MDR with increasing FAR - can be observed. This analysis also shows, that even with a substantially increased FAR, the solutions 1 and 4 still show an infeasible MDR. The MDR of solution 7 does not increase when lowering the FAR below the initial 5%. This suggests that the minimal faulty behavior does not overlap with the nominal behavior at all, and a strict separation of both states is possible. This shows the importance of the analysis of the FAR on the MDR since the solution 7 can achieve better results with a lower FAR.

The evaluation of the isolation is not shown here due to space limitations. To assess the identification performance of the solutions, the two best performing ones with the lowest costs are compared. These are the solutions 3 and 9 as they perform well both for the detection of motor and hydro-mechanical faults as well as for the volumetric pump faults while having the lowest costs in their specific cluster.

To evaluate the identification performance of the chosen solutions they have to be implemented for identification either in the direct or indirect form. As stated in Section 3.4 this evaluation is based on two measures: the rate of correct classification and the average spread of the predicted health inter-

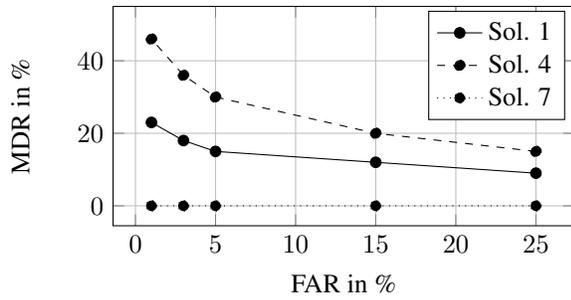


Figure 7. Missed detection rates for the volumetric pump fault for the solutions 1, 4 and 7 based on the FAR used for HDR calculations with a fixed minimum detection requirement of 0.3

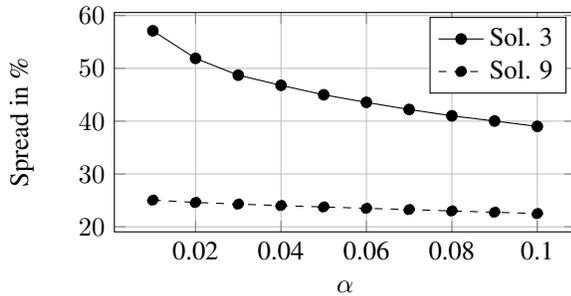


Figure 8. Spread of the identified health grade for the fault $f_{vol,pump}$ for the solutions 3 and 9.

val. Since the spread of the predicted health is only assessable for the direct implementation and comparing directly and indirectly implemented results is a complicated task, only the best directly implementable residuals are compared here.

The results of the identification analysis are depicted in Figure 8. The figure shows the spread over the chosen α for the HDR calculation of the health interval. The rate of correct classification is not shown, since it's - similar to the FAR - directly related to α . Sampling a random system in the specified uncertainty bounds will lead to a correct classification rate of $100(1 - \alpha)\%$. The spread shows a similar behavior as the MDR above. For the solution 9 which measures the flow directly, the results are much more accurate than for the solution 3. The only directly implementable residual for solution 3 uses the differential pressure over the flow restriction dp_{s3} to indirectly measure the flow. These residuals were part of the worst performing cluster in the MDR analysis of the volumetric pump fault, and thus the results observed here are similar. Even when α is increased substantially, the spread of solution 3 is still worse than the best results of solution 9. This comes with the downside of a much higher cost for solution 9 compared to 3 due to the costly volumetric flow sensor.

4.6. Implementation

The following section shows the application of the chosen solutions to real sensor data of the HPP test rig. As explained above, the test rig was not built to design and test FDI schemes, and thus it has no ways to emulate faults of the system. Consequently, the applied solutions 3 and 9 should not detect any faults. In fact, this is the observed behavior. In none of the tested cases, any of the detection residuals detected a fault. This proves that the chosen parameter uncertainty intervals cover the behavior of the HPP test rig, however, it does not prove that faults can be detected and show the modeled behavior. Even though the test rig does not show any faults, one of the conducted analysis is presented in the following section.

One of the tests is shown in Figure 9. The first two panels show the normalized sensor signals for the pump volumetric flow and the angular velocity of the pump respectively. The gray lines show the raw sensor data and the black lines the mean of that signal for the current steady-state identified by the steady-state assessment and aggregated by the aggregation blocks shown in Figure 1. The areas where no mean is shown are considered transient. These sensor signals are used in the fault identification residual of solution 9. It detects volumetric pump flows by comparing the theoretical flow deduced from the speed of the pump with the actual measured flow. The vertical dashed lines show points in time when an evaluation of this MSO would be conducted, given it is triggered by the expert system or implemented as a detection residual. Note that the high flow phase after 40 s is not long enough for the signals to reach steady-state, and thus no residual is triggered.

Since the applied diagnostic engine doesn't detect any faults, the implemented identification residuals are not triggered automatically. To still be able to show results of the fault identification, these residuals are triggered manually to produce the results shown in the last panel of Figure 9. The dark grey bars show the range of the most probable 95 % of fault values for the volumetric pump fault $f_{vol,pump}$ identified by solution 9. Each bar's height marks the identified range and is spread over the entire steady state length. It is apparent that the bars are shallower than predicted in Section 3.4. This is due to the fact, that only the plausible values in $[0, 1]$ are shown. The actual identified interval also include values below 0. This increases the accuracy for the identification of faults near 0. For the analysis around 25 s this effect leads to a particularly shallow interval of $[0, 0.09]$. In fact, for all the shown analysis, the identified interval is smaller for points with higher flow through the system. This comes from the increased volumetric efficiency of pumps for higher flows. The underlying model doesn't model this dependency and uses a nominal efficiency coefficient as shown in Equation 9. To cover all modes of operations with low and high flows, this effi-

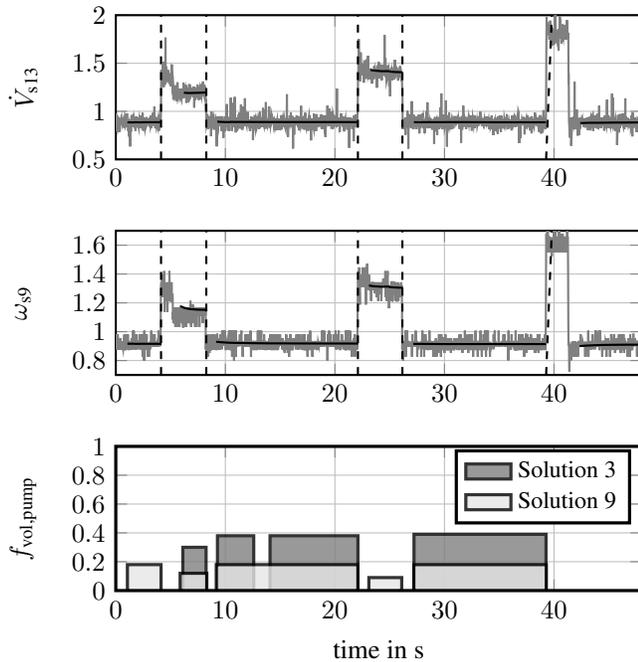


Figure 9. Raw and processed test rig sensor data as well as the identified fault intervals

ciency parameter has to have a rather wide uncertainty. Consequently, when the pump has a high volumetric efficiency in high flow operating modes, the potential fault cannot take on high values, since the parameter is already at its plausible end. This fact increases the accuracy for high flow operating modes and shows that the tested unit probably has a health grade of close to one. This suggests that modeling the flow rate dependent volumetric efficiency explicitly could improve the diagnosis results. For the FDII architecture itself, this suggests that aggregating the health status over multiple evaluation points could lead to overall better results.

The lower panel of Figure 9 also shows results for the evaluation of the identification residual of solution 3 in light gray. The steady state phases and thus points in time when the residual is evaluated differ from the ones for solution 3 since different sensors are used and the steady-state of these sensors is slightly different from the once used before. The general results show the predicted behavior during the evaluation of the results. Solution 3 performs worse than 8 due to the indirect flow measurement. The health grade intervals are generally twice as wide. The effect of narrower intervals for higher flows can be observed as well, but isn't as noticeable since the superimposed uncertainties from the added equations shadow that effect.

5. CONCLUSION

This paper presents a methodology to design robust model-based maintenance focused FDII schemes. The FDII engine itself utilizes the statistical evaluation of model-based residuals using Monte-Carlo simulations and highest density regions. This allows to consider measurement and parameter uncertainties when evaluating residuals, which increases the robustness of the method. The design methodology employs the structural analysis of available design models of the system to propose possible FDII schemes as minimal sensor sets. These sets are subsequently tested for their fault sensitivity to facilitate a sound selection of optimal solutions.

The application of the presented methodology to an aircraft hydraulic power package shows the advantages gained by applying a structured method in an FDII design process. The structural analysis of the behavioral model of the HPP reveals that even with all considered sensors in place, the low pressure filter fault is not detectable and the motor and the hydro-mechanic pump faults are not isolable from each other. Additionally, the structural analysis shows that there are 54 possible sensor sets which fulfill the FDII requirement. Calculating these solutions manually would be a tedious and error-prone task, which is significantly eased by the use of structural analysis.

The subsequent evaluation of all possible sensor sets in terms of costs and robustness reveal their differences in a quantifiable way. This step shows that the combined uncertainties in the measurements and parameters lead to a practically undetectable high pressure filter fault. This illustrates the importance of the evaluation of the results and that structural analysis can produce practically unfeasible solutions. In addition, the evaluation helps to assess trade-offs, as shown by the comparison of the accurate but also expensive direct measurement of the volumetric flow and its indirect counterpart.

The application of selected solutions to a real test rig proves that the modeled nominal behavior matches the real data and the selected solutions do not produce any false alarms. The identified health grades of the real test rig are also in line with the assumed nominal components. Since the test rig is not capable of emulating faults of the system, the diagnostic capabilities cannot be tested entirely. This does not reduce the scope of validation of the method, but rather of the used model's accuracy.

ACKNOWLEDGMENT

This work was funded by the German Federal Ministry for Economic Affairs and Climate Action within projects RTAPHM (contract code: 20X1736M) and MODULAR (contract code: 20Y1910G) in the national LuFo program. Their support is greatly appreciated.



REFERENCES

- Air Transport Association. (2002). *ATA MSG-3 Operator/Manufacturer Scheduled Maintenance Development* (Tech. Rep. No. 2002.1). Air Transport Association.
- Cassar, J., & Staroswiecki, M. (1997). A structural approach for the design of failure detection and identification systems..
- Hyndman, R. J. (1996). Computing and graphing highest density regions. *The American Statistician*, 50(2), 120–126. (Publisher: Taylor & Francis Group)
- Krysander, M., & Frisk, E. (2008). Sensor placement for fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 38(6), 1398–1410. (Publisher: IEEE)
- Mardt, F., & Thielecke, F. (2021, June). Robust Model-Based Fault Detection Using Monte Carlo Methods and Highest Density Regions. In *6th European Conference of the Prognostics and Health Management Society PHM*. PHM Society.
- Rosich, A., Frisk, E., Aslund, J., Sarrate, R., & Nejjari, F. (2011). Fault diagnosis based on causal computations. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 42(2), 371–381. (Publisher: IEEE)
- Trochermann, N., Rave, T., Thielecke, F., & Metzler, D. (2017). An investigation of electro-hydraulic high efficient power package configurations for a more electric aircraft system architecture. In *Deutscher Luft- und Raumfahrtkongress DLRK*.
- Trochermann, N., & Thielecke, F. (2021). Control Strategies for a Dual AC Motor Pump System in Aircraft Hydraulic Power Packages. In *The 17th Scandinavian International Conference on Fluid Power, SICFP'21, May 31- June 2, 2021, Linköping, Sweden*.

Prognosis of wear progression in electrical brakes for aeronautical applications

Andrea De Martin¹, Giovanni Jacazio², Vincenzo Parisi³ and Massimo Sorli⁴

^{1,2,3,4}*Politecnico di Torino, Department of Mechanical and Aerospace Engineering, Torino, 10129, Italy*

*andrea.demartin@polito.it
giovanni.jacazio@formerfaculty.polito.it
vincenzo.paris@studenti.polito.it
massimo.sorli@polito.it*

ABSTRACT

The evolution towards “more electric” aircrafts has seen a decisive push in the last decade, due to the growing environmental concerns and the development of new market segments (flying taxis). Such push interested both the propulsion components and the aircraft systems, with the latter seeing a progressive trend in replacing the traditional solutions based on hydraulic power with electrical or electro-mechanical devices. Although more attention is usually devoted towards the flight control actuation, an interesting and fast-developing application field for electro-mechanical systems is that of the aeronautical brakes. Electro-mechanical brakes, or E-Brakes hereby onwards, would present several advantages over their hydraulic counterparts, mainly related to the avoidance of leakage issues and the simplification of the system architecture. The more difficult heat dissipation, associated with the thermal issues that usually constitute one of the most significant sizing constraints for electro-mechanical actuators, limits so far, their application (or proposal of application) to light-weight vehicles. Within this context, the development of PHM solutions would align with the need for an on-line monitoring of a relatively unproven component. This paper deals with the preliminary stages of the development of such PHM system for an E-Brake to be employed on a future executive class aircraft, where the brake is actuated through four electro-mechanical actuators. Since literature on fault diagnosis and prognosis for electrical motors is fairly extensive, we focused this preliminary analysis on the development of PHM techniques suitable to monitor and prognose the evolution of the brake pads wear instead. The paper opens detailing the system architecture and continues presenting the high-fidelity dynamic model used to build synthetic data-sets representative of the possible operating conditions faced by the E-Brake within realistic operative scenarios. Such data are then used to foster a

preliminary feature selection process, where physics-based indexes are compared and evaluated. Simulated degradation histories are then used to test the application of data-driven fault detection algorithm and the possible application of particle-filtering routines for prognosis.

1. INTRODUCTION

Despite being a rather recent subject, the development of E-Brake systems is a critical step towards the complete electrification of aircraft systems. Given their relatively unproven technology, the definition of a comprehensive PHM system would provide additional confidence towards their application, lowering the risk of unanticipated failures, reducing the aircraft downtimes and giving access to strategic information useful to optimize the fleet management. Although literature on PHM activities for the most common components of electro-mechanical brakes is extensive, few papers have been published about the E-Brakes themselves. In (Ramesh et al. 2021) authors propose a FDI algorithm to observe and correctly assess the most probable failures occurring in a simple electro-mechanical brake for aeronautic applications. The analysis considers an aeronautical brake actuated by one Electro-mechanical actuator driven through a brushed DC-motor and is mainly focused on electrical failures. In (Oikonomou et al. 2022) authors investigate the prognosis of wear in aeronautical brakes through the analysis of historical series of brake pads thickness. Data-driven techniques are applied to perform the long-term prognosis, and the results of an interesting benchmarking activities comparing the performances of several algorithms are provided. Results are promising but assume the presence of dedicated sensors to measure the thickness of the brake pads, which are not foreseen for the application under study in this paper. The E-LISA research project, under way within the Clean Sky 2/Clean Aviation framework, has the objective of developing an innovative iron bird dedicated to executing tests on the landing gear of a small aircraft equipped with an electro-mechanical landing gear and electrical brake. The E-

Andrea De Martin et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

LISA iron bird consists of a multi-functional intelligent test facility integrating hardware and software, allowing all the tests and analyses perceived as fundamental to be performed to demonstrate the maturity of an electro-mechanical landing gear, hence paving the way for its implementation in a small passenger aircraft and will include prognostics and health management (PHM) functionalities for the electrical brake system. Such tests include the simulation of complete landing procedures under different operating conditions such as runway friction (wet/dry), presence of waving and irregularities along the runway, variable aircraft weight, and approach speed as detailed in (De Martin et al. 2022). This paper deals with the preliminary analysis needed to develop a comprehensive PHM system for the E-Brake, focusing on the feature selection, fault detection and prognosis of wear in the brake pads without the addition of dedicated sensors considering the case study of a real E-Brake system, currently under development. To meet such objective authors resorted to the definition of a high-fidelity simulation environment able to represent the expected behavior of the electrical brake under variable operating conditions and dynamically evolving wear conditions, mirroring the approach already applied to primary flight control actuators in (Autin et al. 2021). The paper is organized as follows. At first the case study under consideration is presented in detail, highlighting its most prominent characteristics and their expected effect on the definition of the prognostic system. The operational scenario considered for the E-Brake at hand is also introduced and discussed. The paper focus is then shifted to the high-fidelity simulation model and the definition of a suitable degradation model to replicate the effects of wear progression on the brake behavior. Simulation results are then used to pursue the feature selection process and used to foster the definition of a fault detection/failure prognosis framework based upon a combination of data-driven and particle filtering techniques. Such algorithms are then applied to the most suitable feature candidates and their results compared.

2. CASE STUDY AND OPERATIONAL SCENARIO

The case study under analysis is an E-Brake system for an executive-class aircraft with an expected weight at take-off of around 6 tons. Two E-Brake systems are integral with the Main Landing Gear system, one for the Left-Hand side, one

for the Right-Side each. As depicted in Figure 1, each E-Brake is a multi-disk assembly actuated through four Electro-Mechanical Actuators (EMAs) controlled in force. Whenever the pilot acts on the brake pedals, a force command is sent towards the E-Brake system; such command signal is processed by the Brakes Control Unit (BCU), which can cut the force command signal through the touch-down protection routines, avoiding that the brakes are actuated before the aircraft rotation during landing has ended. The command signal can be further modulated by the electronic anti-skid system, which decreases the force request depending on the runway conditions to avoid the occurrence of wheel blockage events and excessive slip according to a combination of pilot input and automatic recognition of the runway status.

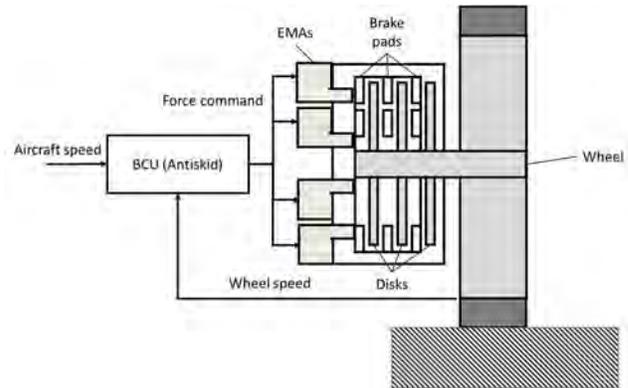


Figure 1. Case-study architecture.

The electro-mechanical actuators, which schematics is provided in Figure 2, are driven by one Brushless-DC motor with each, and act on the brake pads through a mechanical transmission made of a one-stage reducer and a ball-screw. Each actuator is equipped with a force sensor to measure the exerted action, while a resolver integral with the motor shaft is used to infer its position and realize the Field Oriented Control of its phase currents. The PHM system considered for this research activity is projected to work at the Landing Gear System level, thus having access to all the signals available within the Landing Gear control system, including:

- E-Brake motors phase currents
- Angular position of the E-Brake motors shaft
- Pilot command

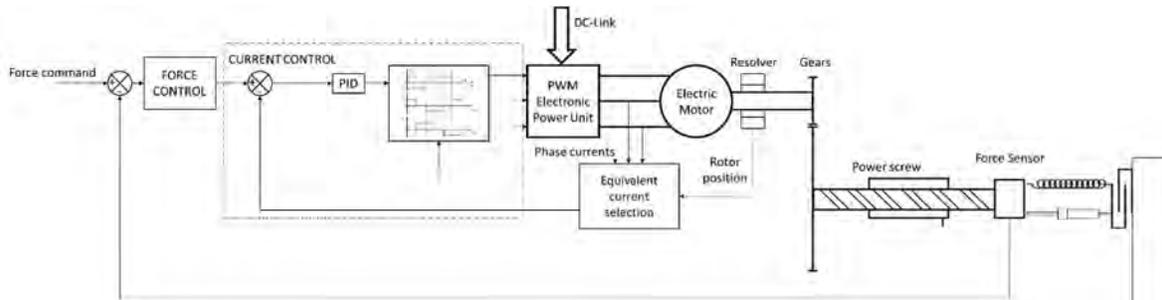


Figure 2. The E-Brake actuators.

- Anti-skid and touch-down protection signals
- Exerted force measurement for each EMA

To complete the description of the case study under analysis it is also critical to introduce the operational scenario faced by the landing gear system during its projected functional life. Such operation is important both to properly characterize the expected behavior of the E-Brake system under nominal health status, but also to anticipate the effects that significant variations in the operational conditions may have on the E-Brake signals and thus on the possible feature candidates. Starting with the operating temperature, the E-Brake system is expected to operate in presence of external temperatures ranging from -40°C to $+60^{\circ}\text{C}$ in combination with variable runway conditions (dry, wet, snowy, with ice). Moreover, a variation of $\pm 10\%$ of the average weight at landing can be expected as a function of the fuel consumption and passengers' number. The same percentual variation is also applied on top of the nominal aircraft horizontal speed during the final approach (50 m/s).

3. SIMULATION MODEL

To replicate the Landing Gear system behavior, we resort to the high-fidelity model provided in (De Martin et al. 2022), capable of representing the entire landing procedure under the assumption of symmetric touch-down, thus with null roll angle and equal weight repartition between the two main landing gear legs. In such model the aircraft is represented as a rigid body subjected to the aerodynamic components of Lift due to the aircraft wing (L), anterior wing (L_{fw}), tail wing (L_t), and Drag (D). Each of these components is expressed as a function of the attack angle of the related aerodynamic surface and scales with the square of the wind speed. Making reference to Figure 3, the vertical, horizontal and pitch dynamics of the vehicle can then be represented as a system of three differential equations.

$$\begin{cases} -2F_{MLG} - F_{NLG} - W + L \cos\vartheta + L_t \cos\vartheta + L_{fw} \cos\vartheta + T \sin\vartheta - D \sin\vartheta = m_{air} \ddot{z} \\ 2F_{MLG,i} H_{MLG} - L H_L - L_t H_{L_t} + L_{fw} H_{L_{fw}} - F_{NLG} H_{NLG} = I_{yy} \ddot{\vartheta} \\ T \cos\vartheta - \sum_i F_{t,i} = m_{tot} \ddot{x} \end{cases} \quad (1)$$

where $\cos\vartheta = \cos\vartheta$ and $\sin\vartheta = \sin\vartheta$. T is the motors thrust, while F_{MLG} , F_{NLG} and $F_{t,i}$, are respectively the force exerted on the aircraft by the Main Landing Gears, the Nose Landing Gear and the friction forces between the aircraft tires and the runway (Cook 2012). The main inertia moment I_{yy} and the position of the center of mass of the aircraft (thus the torque lever arms H_D, H_L, H_{L_t}, \dots) are variable according to data provided by the industrial partners of the project as a function of the aircraft acceleration and attitude. The total mass of the aircraft is addressed as m_{tot} equal to the sum of the aircraft mass (m_{air}) and the overall mass of the main and nose landing gears.

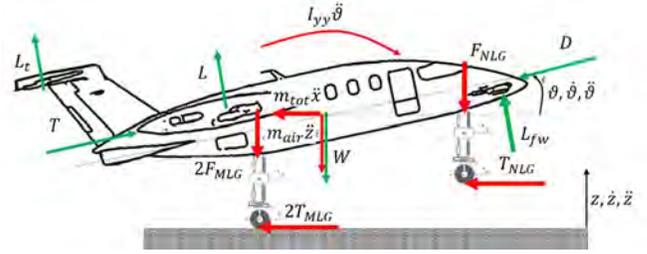


Figure 3. Aircraft dynamic equilibrium.

Each landing gear leg is described as a two-degrees-of-freedom system as shown in Figure 4. The shock absorber characteristics are known and provided by the landing gear supplier. Similarly, the vertical dynamics of the leg can be represented as a function of the stiffness (k_t) and damping (c_t) coefficient representative of the contact between tire and runway, while x_w and x_{rw} are respectively the vertical displacement of the wheel and that of the runway simulator, by default equal to zero. The wheel mass is m_w , while m_{leg} is the mass of the leg. The characteristics of the tire are derived from data provided by the industrial partners.

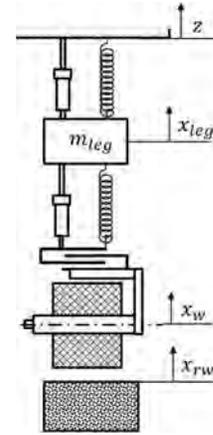


Figure 4. Landing Gear leg model.

The rotational dynamics of the wheel can be described according to the free-body diagram of Figure 5. We address with $F_n = k_t(x_w - x_{rw}) + c_t(\dot{x}_w - \dot{x}_{rw})$ the vertical force exchanged between the wheel and the runway and with $F_t = F_n \mu$ the friction force. u_{rw} is the rolling friction parameter, expressed as a function of the wheel angular frequency and of the tire pressure (Carbone and Putignano 2013).

$$\begin{aligned} F_n \mu \left[\frac{D_w}{2} - (x_{leg} - x_{rw}) \right] \text{sign}(\lambda) - F_n u \tanh \vartheta_w \\ - c_w \dot{\vartheta}_w - T_{brk} = I_w \ddot{\vartheta}_w \end{aligned} \quad (2)$$

where ϑ_w is the wheel rotation, I_w is the moment of inertia of the wheel assembly, D_w its diameter and c_w the viscous friction coefficient roughly representative of the dissipation in the wheel supports. The friction coefficient μ is evaluated according to the Burckhardt model (M. Burckhardt 1993) as

a function of the slip factor λ between wheel and runway simulator.

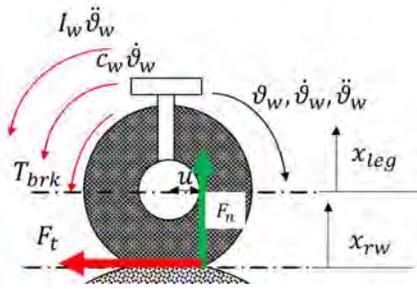


Figure 5. Aircraft dynamic equilibrium.

The E-Brake system is made of four Electro-Mechanical Actuators (EMAs) controlled in force and acting in parallel on a disk brake. The EMAs model is derived from the dynamic representation of similar systems employed as flight control actuators (De Martin et al. 2017). The control system is described as a two-nested control loops, where a sequence of Proportional-Integrative controllers operates on the force control loop and on the current control loop of each brushless motor. The sensors are modelled through second order transfer functions replicating the expected dynamics of the load cell and of the Hall-effect sensors employed to monitor the angular position of the Brushless-DC rotor. The simulation of the measure chain is complete with the model of the employed A/D converters. The dynamic model of each EMA features a functional description of the Electronic Power Converter derived from (Mohan et al. 2005) for a three-phase inverter controlled through Pulse Width Modulation (PWM). The electrical dynamics of the motor is described according to a streamlined three-phase model of the system, where $V_{a,b,c}$ and $i_{a,b,c}$ are the phase voltages and currents.

$$\begin{aligned} [V_{a,b,c}] &= [R_{a,b,c}(T_w)][i_{a,b,c}] + \\ [L(T_w)] \frac{d}{dt} [i_{a,b,c}] &+ \frac{d}{dt} [\phi_{a,b,c}(\vartheta_{el})] \end{aligned} \quad (3)$$

$[R_{a,b,c}]$ is the electric resistance matrix, which elements depends on the windings' temperature (T_w). $[L]$ is the inductance matrix, accounting for self-induction and mutual induction phenomena along with the effect of magnetic flux dispersion. Finally, $[\phi_{a,b,c}]$ is the concatenated magnetic flux provided by the permanent magnets, function of the electrical angle (ϑ_{el}). The torque at the motor shaft can then be computed, leading to the the dynamic equilibrium of the rotor

$$\begin{aligned} \sum_{a,b,c} \frac{d\phi}{dt} i_{a,b,c} - c\dot{\vartheta}_m - k_m(\vartheta_m - \vartheta_{gb}) \\ - c_m(\dot{\vartheta}_m - \dot{\vartheta}_{gb}) = I_m \ddot{\vartheta}_m \end{aligned} \quad (4)$$

where ϑ_m and ϑ_{gb} are the angular position of the motor shaft and of the gears. I_m is the moment of inertia of the rotor, while k_m and c_m address the torsional stiffness of the motor shaft and its associated damping. The gear pair is described

as a rotational mass-spring-damper system, thus leading to the following equation,

$$\begin{aligned} k_m(\vartheta_m - \vartheta_{gb}) + c_m(\dot{\vartheta}_m - \dot{\vartheta}_{gb}) \\ - \frac{1}{\tau} [k_{gb}(\vartheta_{gb} - \vartheta_{rs}) + c_{gb}(\dot{\vartheta}_{gb} - \dot{\vartheta}_{rs})] - T_{fr,gb} \\ = I_{gb} \ddot{\vartheta}_{gb} \end{aligned} \quad (5)$$

where τ is the transmission ratio, $T_{fr,gb}$ the friction torque, while ϑ_{rs} is the angular position of the rotating part of the screw. The friction torque is computed as the sum of three components, one dependent on the acting load, one related to the viscous friction and a drag torque component. The power-screw is modelled as a two-degrees of freedom elements, where the rotating part is connected to the translating element through a viscoelastic element. Defining with $x_{rs,i}$ the position of the translating portion of the screw pertaining to the i -th actuator, it becomes possible to describe the brake dynamics, and thus that of the pads. Addressing with k_{eb} the stiffness, it is possible to evaluate the braking torque acting on the landing gear wheel as a function of the translating mass of the brake pads m_{eb} , its translation x_{eb} and the angular speed of the wheel $\dot{\vartheta}_w$ as,

$$\begin{cases} T_{brk} = 0 \leftrightarrow x_{eb} < x_{thr} \\ T_{brk} = f_{eb}[k_{eb}(x_{eb} - x_{thr}) - c_{eb}(\dot{x}_{eb})] \leftrightarrow x_{eb} \geq x_{thr} \end{cases} \quad (6)$$

where $f_{eb} = f_{eb}(\dot{\vartheta}_w)$ is the friction coefficient between the brake pads and disk, function of the wheel angular frequency. Knowing the braking torque and the wheel angular frequency it is possible to compute the mechanical power transformed into heat by the braking process. Such power is used within a simplified thermal model of the E-Brake assembly to estimate at each time step the temperature of the pads and the temperature of the electric motor windings considering both the thermal power generated by the motor themselves and that transmitted to the external environment. Since the pads contact the brake disks only when their translation x_{eb} overcomes a predefined stroke equal to x_{thr} , it is possible to model the effects of the pads wear by properly increasing such threshold value under the assumption that the brake pads return in the original position once the braking procedure is finished. According to (Olesiak et al. 1997; Yevtushenko et al. 2017), wear progression in brake pads can be described as dependent on an experimental coefficient f_{wear} and k_{wear} , function of the local absolute temperature T , the sliding velocity between disks and pads v , and the contact pressure p .

$$\Delta x_{thr} = \int_t f_{wear}(T) K_{wear}(T) v(t) p(t) dt \quad (7)$$

Expressing the sliding velocity as a function of the wheel angular frequency $\dot{\vartheta}_w$ and the radial coordinate of the pads with respect to the wheel axis R_{pad} , we have

$$v = \dot{\theta}_w R_{pad} \tag{8}$$

The average pressure within the pads/disks contact area can be computed as a function of the braking force exerted by the four actuators and the pad contact area.

$$p = \frac{k_{eb}(x_{eb} - x_{thr}) - c_{eb}(\dot{x}_{eb})}{A_{pad}} \tag{9}$$

An example of the model behavior for different level of wear (no wear and critically advanced wear), is provided in Figure 6, where it is evident how the pads wear introduces an additional delay on the brake response due to the additional stroke required to cover the additional gap.

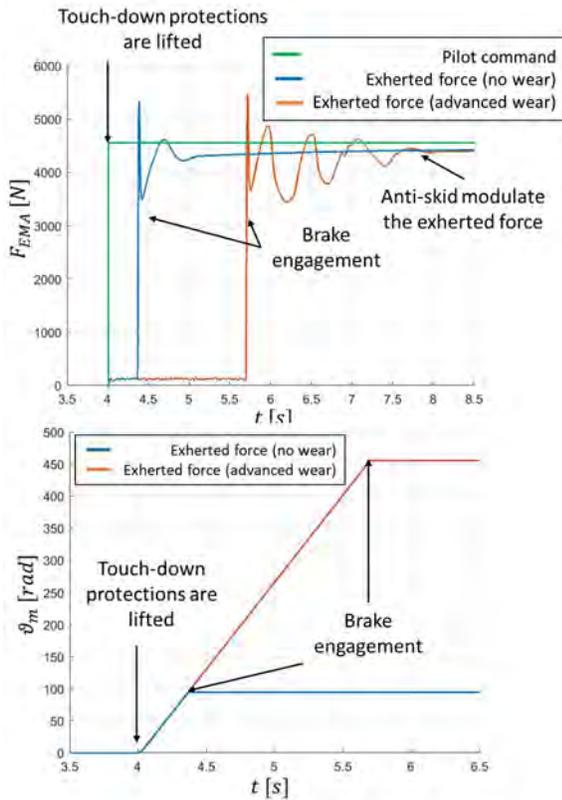


Figure 6. Model response in nominal and severely degraded conditions.

4. FEATURE SELECTION

The high-fidelity model was employed to build a preliminary database considering increasing levels of wear, varying the operating conditions (temperature, runway conditions, approach speed) as indicated in Section 3. Moreover, Gaussian noise was applied on the parameters of each of the four EMAs driving the E-Brake. The characteristics of such noise are dependent on the nature of the perturbed parameters and are meant to represent the effects of the expected production and design tolerances on the EMA behavior. As a consequence, the behavior of each EMA is slightly different from the others, even though the nominal

value of the parameters used to model their dynamics is the same. Given the time-consuming nature of simulating an entire landing process, a total of thirty simulations for each considered wear level were performed. As anticipated in Section 3, the main macroscopic effect of the wear progression is in the increase of the free stroke that needs to be covered by the EMAs to get the pads in touch with the disks, leading to the progressive increase of the delay between the force command sent to the actuators and the actual generation of a braking torque. Considering this observation and given the signals considered available to the PHM system, a few feature candidates were isolated and evaluated according to correlation, accuracy, signal-to-noise ratio as advised by (Vachtsevanos et al. 2006). The most prominent feature candidates analyzed in this work are reported in Table 1 along with their performance indexes, while their behavior is summarized in Figure 7. Please notice that to avoid excessive clutter within the figure, the scatter plots report only the points correspondent to the average of the data distribution obtained for each considered level of wear. Moreover, features are reported as non-dimensional in order not to disclose sensible data on the E-Brake under examination. The first feature candidate, f_1 , is computed as the delay between the time instant at which the motor shaft begins to move $t_{0,m}$ and the time instant for which the braking torque signal raises from zero $t_{0,f}$. Both time instants are isolated studying the moving variance of the motor shaft position signal and the braking force signal. The second feature candidate f_2 is instead computed as the angular rotation of the motor shaft $\Delta\theta_m$ between the time instants $t_{0,m}$ and $t_{0,f}$. Although these two candidates are expected to be correlated, they are expected to be affected differently from the variations in the operating conditions. The second feature is expected to be less influenced by the fluctuations in the mechanical efficiency of the EMA due to low temperatures or lubrication ageing, making it the preferable choice between the two in the considered case. This second feature might be however not suitable if the resolution of the motor shaft position signal is lacking. The third considered feature candidate f_3 is the autocorrelation between the force signal measured during the braking procedure and a database of the expected braking force during landing.

Table 1. Feature candidates

Feature	Correlation	Accuracy	S/N Ratio
$f_1 = \Delta t$	0.98	0.98	7.65
$f_2 = \Delta\theta_m$	0.99	0.9	7.98
$f_3 = \rho(F_{EMA})$	0.96	0.75	3.65

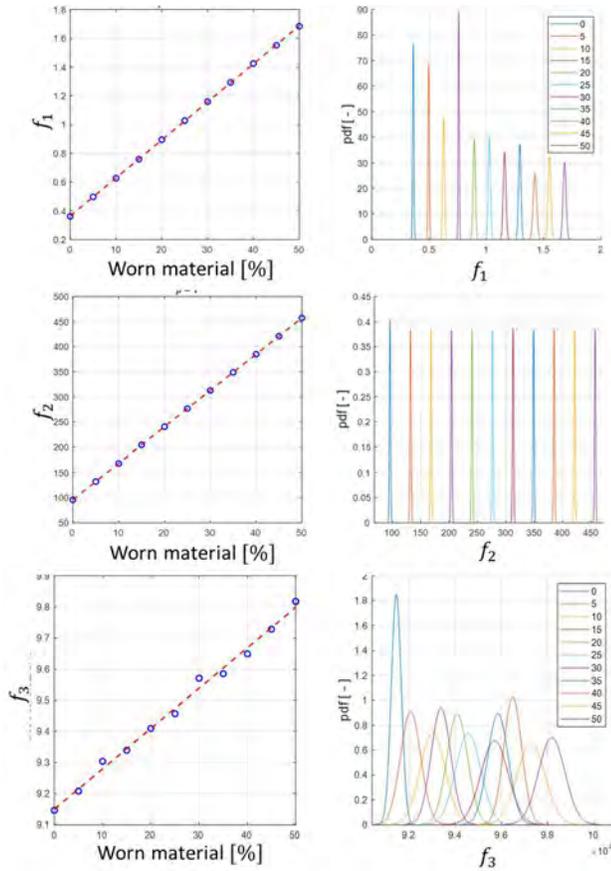


Figure 7. Feature candidates' behavior.

This feature provides interesting results and would be advantageous over the first two options since it does not require the knowledge of the angular position of the motor shaft. It is however expected that other failure modes, related to the motor electrical windings, can affect its behavior and has been as such discarded.

5. FAULT DETECTION

The chosen fault detection routine is based on a simple data-driven technique, where the running distributions of the selected features are compared against baselines representative of the nominal health behavior. In this scheme, the fault alarm is raised if the confidence level in the fault declaration overcomes a pre-defined threshold. According to (Vachtsevanos et al. 2006), such approach allows to adapt the baseline conditions to the peculiarities of each monitored subsystem, containing at the same time the computational effort required to process the signals coming from the field. To evaluate the behavior of the proposed routine and the performances of the proposed features, we simulated the possible life cycle of three E-Brake systems through the high-fidelity model presented in Section 3. The model was deployed to simulate a succession of landing in randomly varying operating conditions, computing at each time step the wear progression according to the model described by

Equations (7-9). The tests were performed considering a confidence level threshold of 0.95. Example of the obtained results are reported in Figure 8, where the features distributions for healthy (in white), current (in yellow or grey) and at fault declaration (in red) conditions are depicted for each fault type. The system also provides in output the expected confidence associated with the fault declaration and the eventual presence of a fault alarm. The fault detection routine was able to successfully detect the early insurgence of brake pads wear in all the presented test cases. The average time at detection for the considered features is estimated in 186 landing events for f_2 and 386 landing events for f_1 .

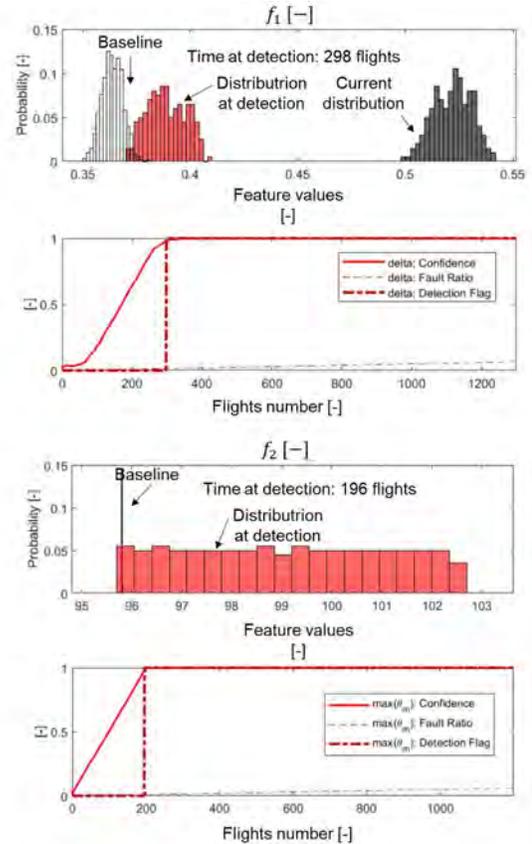


Figure 8. Examples of fault detection results for features f_1, f_2 .

6. PROGNOSIS

Prognosis is achieved through a Bayesian estimation method using a particle filtering approach as firstly proposed by (Orchard and Vachtsevanos 2009). This method takes advantage of a nonlinear process (fault / degradation) model to describe the expected dynamics of the fault progression and a measure model derived from the feature/wear progression dependence observed during the feature selection phase. Prognosis through particle filtering is achieved by performing two sequential steps, prediction and filtering. Prediction uses both the knowledge of the previous state

estimate and the process model to generate the a priori estimate of the state probability density functions (pdfs) for the next time instant,

$$p(x_{0:t}|y_{1:t-1}) = \int p(x_t|y_{t-1})p(x_{0:t-1}|y_{1:t-1}) dx_{0:t-1} \quad (10)$$

This expression usually does not have an analytical solution, requiring Sequential Monte Carlo algorithms to be solved in real-time with efficient sampling strategies (Roemer et al. 2011). Particle filtering approximates the state pdf using samples or “particles” having associated discrete probability masses (often called “weights”) as,

$$p(x_t|y_{1:t}) \approx \tilde{w}_t(x_{0:t}^i)\delta(x_{0:t} - x_{0:t}^i)dx_{0:t-1} \quad (11)$$

where $x_{0:t}^i$ is the state trajectory and $y_{1:t}$ are the measurements up to time t. The simplest implementation of this algorithm, the Sequential Importance Re-sampling (SIR) particle filter (Arulampalam et al. 2009), updates the weights using the likelihood of y_t as:

$$w_t = w_{t-1}p(y_t|x_t) \quad (12)$$

Although this traditional particle filtering technique has limitations, in particular with regards to the description of the distributions tails, and more advanced resampling schemes have been proposed (Acuña and Orchard 2017), this technique was still deemed valid for a purely preliminary analysis. Long-term prediction of the fault evolution can be obtained by iterating the “prediction” stage, and are used to estimate the probability of failure in a system given a hazard zone that is defined via a probability density function with lower and upper bounds for the domain of the random variable, denoted as H_{lb} and H_{up} , respectively. Given the probability of failure, the RUL distribution for any given prediction can be computed along with the risk function (Acuña and Orchard 2018).

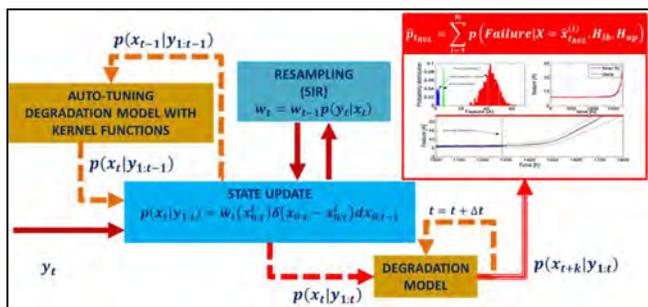


Figure 9. Prognostic routine.

The algorithm adopted for this paper follows the scheme provided in in Figure 9 (De Martin et al. 2018); this approach makes use of degradation models that are tuned or their parameters adjusted through Recursive Least Square (RLS) algorithm embedded in the main routine, to compute the current a priori state of the system, $p(x_t|y_{1:t-1})$, and to

perform the iterative calculation that leads to the long term prediction $p(x_{t+k}|y_{1:t})$. Auto-tuned models are required to describe and follow changes in the degradation process and to describe the process and measurement noise.

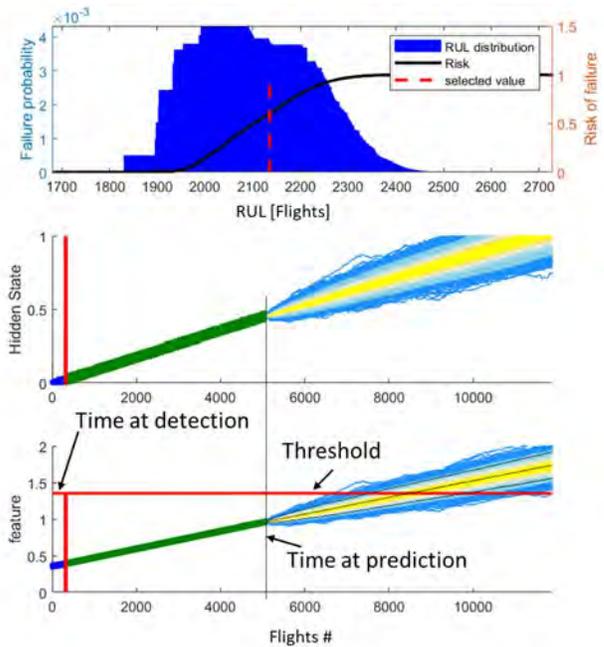


Figure 10. Prognostic output of the particle filtering framework for feature f_1 .

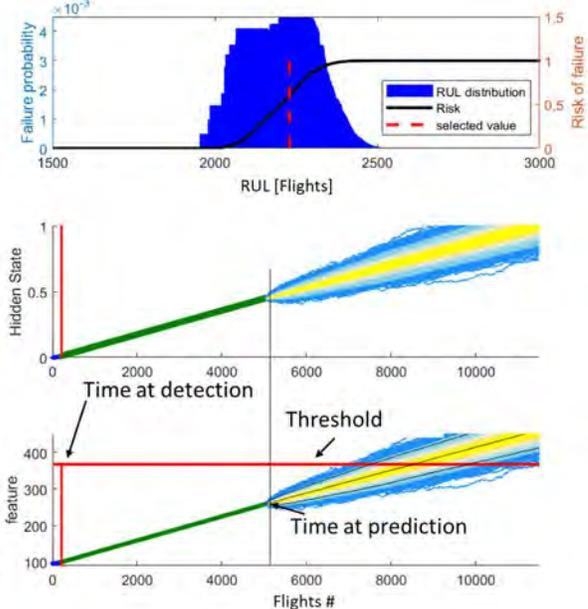


Figure 11. Prognostic output of the particle filtering framework for feature f_2 .

An example of the results, comparing the behavior of the same algorithm applied to the two feature candidates f_1 , and

f_2 , are depicted in Figures 10 and 11, considering the same time at prediction. This example shows anecdotal evidence, further supported by the performance analysis provided in the next part of the paper, of the more accurate predictions provided by the feature f_2 . The prognostic outputs were then analyzed through metrics traditionally employed in the preliminary analysis of PHM systems like the mean relative accuracy (RA) and the definition of a prognostic horizon (PH) associated with a certain relative accuracy threshold (Saxena et al. 2008). Although more rigorous metrics can be found in literature, mainly assessing the capability of the prognostic routine of estimating the probability distribution of the real Remaining Useful Life of the component, the presented study is focused on a first assessment of the feasibility and the benefit of a prognostic system for this particular application and not on the research for a new, more accurate technique for prognosis, which is planned for later stages of the research program. The $\alpha - \lambda$ diagram for the less favorable degradation history is provided Figure 12 for both features, highlighting the convergence of the RUL prediction towards the ground-truth and a stable accuracy level. In the considered case, the Prognostic Horizon accounts for the 68.7% of the projected operative life of the pads for the predictions performed with f_1 , raising to more than 77.5% for analysis provided through the second feature f_2 . The average RA during the performed simulations is above 80% in both cases.

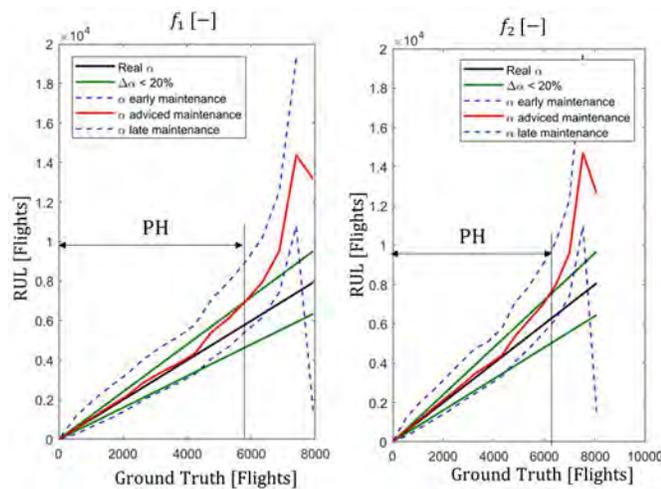


Figure 12. $\alpha - \lambda$ diagrams of long-term predictions obtained for features f_1 and f_2 .

7. CONCLUSIONS

One of the main aims of the E-Lisa project is to propose the definition of a novel PHM system for Electro-mechanical brakes for aeronautic applications, which are expected to progressively replace the use of traditional hydraulic systems in aeronautic applications. Wear of the brake pads was assessed as one of the most significant failure modes, since it is inevitable during operations. Moreover, it naturally tends

to evolve gradually in time, thus representing a favorable case-study for prognostic applications. To pursue this objective a high-fidelity model of the aircraft, of the landing gear leg and of the E-Brake system was prepared and presented. The model is able to reproduce an entire landing process and incorporate a dynamic degradation model, which simulate wear progression as a function of the simulated operations. Such model was used to generate data necessary to study the main macroscopic symptoms of wear progression and provide the basis for the feature selection process. A few feature candidates have been presented, and their performances evaluated according to traditional metrics. Finally, a fault detection/prognostic algorithm based on a combination of data driven and particle filtering techniques was presented and applied to the simulated datasets. Early results are encouraging, showing the system capability to detect the wear process at its onset and achieving good performance marks for long-term prognosis. Although promising, results shown in this paper are still preliminary and further evaluations are needed. Further work will be in particular addressed at investigating other major failure modes possibly affecting the E-Brake assembly, the impact of such failure modes on the selected features, as well as the definition of a complete Fault Detection and Identification routine.

ACKNOWLEDGMENTS

The research work presented in this paper was performed within the E-Lisa project, which has received funding from the Clean Sky 2 Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement number 887222.

REFERENCES

- Acuña, D. E., and Orchard, M. E. (2017). "Particle-filtering-based failure prognosis via sigma-points: Application to Lithium-Ion battery State-of-Charge monitoring." *Mechanical Systems and Signal Processing*.
- Acuña, D. E., and Orchard, M. E. (2018). "A theoretically rigorous approach to failure prognosis." *Proceedings of the 10th Annual Conference of the Prognostics and Health Management Society 2018 (PHM18), Philadelphia, PA, September 24-27*.
- Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2009). "A Tutorial on Particle Filters for Online Nonlinear/NonGaussian Bayesian Tracking." *Bayesian Bounds for Parameter Estimation and Nonlinear Filtering/Tracking*, IEEE.
- Autin, S., De Martin, A., Jacazio, G., Socheleau, J., and Vachtsevanos, G. J. (2021). "Results of a feasibility study of a Prognostic System for Electro-Hydraulic Flight Control Actuators." *International Journal of Prognostics and Health Management*, 12(3), 1–18.

- Carbone, G., and Putignano, C. (2013). “A novel methodology to predict sliding and rolling friction of viscoelastic materials: Theory and experiments.” *Journal of the Mechanics and Physics of Solids*, Elsevier, 61(8), 1822–1834.
- Cook, M. (2012). *Flight dynamics principles: a linear systems approach to aircraft stability and control*. Butterworth-Heinemann.
- M. Burckhardt. (1993). “Radschlupf-Regelsysteme.” *Fahrwerktechnik: Würzburg: Vogel Verlag*.
- De Martin, A., Jacazio, G., and Sorli, M. (2018). “Enhanced Particle Filter framework for improved prognosis of electro-mechanical flight controls actuators.” *PHM Society European Conference, PHME 2018, Utrecht, Netherlands, July 3-6*.
- De Martin, A., Jacazio, G., and Sorli, M. (2022). “Simulation of Runway Irregularities in a Novel Test Rig for Fully Electrical Landing Gear Systems.” *Aerospace*, 9(2), 114.
- De Martin, A., Jacazio, G., and Vachtsevanos, G. (2017). “Windings fault detection and prognosis in electro-mechanical flight control actuators operating in active-active configuration.” *International Journal of Prognostics and Health Management*, 8(2).
- Mohan, N., Undeland T.M., and Robbins, W. P. (2005). *Power Electronics*. John Wiley and Sons, Inc.
- Oikonomou, A., Eleftheroglou, N., Freeman, F., Loutas, T., and Zarouchas, D. (2022). “Remaining Useful Life Prognosis of Aircraft Brakes.” *International Journal of Prognostics and Health Management*, 13(1), 1–11.
- Olesiak, Z., Pyryev, Y., and Yevtushenko, A. (1997). “Determination of temperature and wear during braking.” *Wear*, 210(1–2), 120–126.
- Orchard, M. E., and Vachtsevanos, G. J. (2009). “A particle-filtering approach for on-line fault diagnosis and failure prognosis.” *Transactions of the Institute of Measurement and Control*.
- Ramesh, G., Garza, P., and Perinpanayagam, S. (2021). “Digital simulation and identification of faults with neural network reasoners in brushed actuators employed in an e-brake system.” *Applied Sciences (Switzerland)*, 11(19).
- Roemer, M. J., Byington, C. S., Kacprzynski, G. J., Vachtsevanos, G., and Goebel, K. (2011). “Prognostics.” *System Health Management: With Aerospace Applications*.
- Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., and Schwabacher, M. (2008). “Metrics for evaluating performance of prognostic techniques.” *2008 International Conference on Prognostics and Health Management*, IEEE, Denver, CO, 1–17.
- Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A., and Wu, B. (2006). *Intelligent Fault Diagnosis and Prognosis for Engineering Systems. Intelligent Fault Diagnosis and Prognosis for Engineering Systems*, John Wiley & Sons, Inc., Hoboken, NJ, USA.
- Yevtushenko, A., Kuciej, M., and Topczewska, K. (2017). “Analytical model for investigation of the effect of friction power on temperature in the disk brake.” *Advances in Mechanical Engineering*, 9(12), 1–12.

Domain knowledge informed unsupervised fault detection for rolling element bearings

Douw Marx^{1,2} and Konstantinos Gryllias^{1,2}

¹ *Division Mecha(tro)nic System Dynamics, Department of Mechanical Engineering, KU Leuven*
douw.marx@kuleuven.be
konstantinos.gryllias@kuleuven.be

² *Dynamics of Mechanical and Mechatronic Systems, Flanders Make*

ABSTRACT

Early and accurate detection of rolling element bearing faults in rotating machinery is important for minimizing production downtime and reducing unnecessary preventative maintenance. Several fault detection methods based on signal processing and machine learning methods have been proposed. Particularly, supervised, data-driven approaches have proved to be very effective for fault detection and diagnostics of rolling element bearings. However, supervised methods rely heavily on the availability of failure data with volume, variety and veracity, which is mostly unavailable in industry. As an alternative data-driven strategy, unsupervised methods are trained on healthy data only and do not require any failure data.

In contrast to supervised and un-supervised data-driven models, physics-based and phenomenological models are based on domain knowledge and not on historical data. Although these models are useful for studying the way in which damage is expected to manifest in a measured signal, they are difficult to calibrate and often lack the fidelity required to model reality. In this paper, an unsupervised data-driven anomaly detection method that exploits informative domain knowledge is proposed. Hereby, the versatility of unsupervised data-driven methods are combined with domain knowledge.

In this approach, supplementary training data is generated by augmenting healthy data towards its possible future faulty state based on the characteristic bearing fault frequencies. Both healthy and augmented squared envelope spectrum data is used to train an autoencoder model that includes regularisation designed to constrain the latent features at the autoencoder bottleneck. Regularisation in the autoencoder loss enforces that the expected deviation of the healthy latent representation towards the augmented latent representation at dam-

aged conditions, is constrained to be maximally different for different fault modes. Consequently, the likelihood of a new test sample being healthy can be evaluated based on the projection of the sample onto an expected failure direction in the latent representation.

A phenomenological and experimental dataset is used to demonstrate that the addition of augmented training data and a specialized autoencoder loss function can create a separable latent representation that can be used to generate interpretable health indicators.

1. INTRODUCTION

1.1. Background on condition monitoring approaches

Condition-based maintenance procedures can help machines operate reliably and continuously by reducing unnecessary maintenance procedures (Lei et al., 2018), and minimizing machine downtime (Lee et al., 2014). Rolling element bearings act as a main source of faults in rotating machinery (Cerrada et al., 2018), and have consequently drawn significant research attention in the condition-based maintenance community. Although impressive fault detection and classification results have been attained using data-driven methods for fault detection in bearings (See Hoang & Kang (2019)), a large majority of these approaches are based on sophisticated supervised learning techniques that require failure data during training.

In contrast, many physics-based, signal processing and unsupervised methods attempt to solve the bearing fault detection problem without the requirement of any fault data. However, these methods bring other challenges. Signal processing methods (Randall & Antoni, 2011) are robust, simple and effective, but often require an expert to interpret the results. Physics-based methods (Cao et al., 2018) and phenomenological (McFadden & Smith, 1984) methods are difficult to design, calibrate and lack the flexibility required to model re-

Douw Marx et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ality. Finally, unsupervised methods lack interpretability and can identify non-fault-related anomalies as machine faults.

To address the respective limitations of these approaches, hybrid methods have been proposed for diagnostics (Leturiondo et al., 2017) and prognostics (Liao & Kottig, 2014) in order to combine the benefits of physics-based, data-driven, domain knowledge and/or signal processing methods. For example, researchers have designed physics-inspired filters for convolutional neural networks (CNNs) (Sadoughi & Hu, 2019). In this work, the parameters of the convolutional kernels are chosen such that respective kernels are sensitive to different bearing faults manifesting in the envelope signal. Others (Liu et al., 2020) have used transfer learning approaches to learn domain invariant features between measured data, and simulated data, allowing improved remaining useful life prediction and a reduced reliance on real world data. Physics-based knowledge was also included in a CNN (Shen et al., 2021) by adding a penalisation to the network loss if predictions are made that are not compatible with expected bearing fault behaviour.

In the context of bearing fault detection, it could even be argued that the use of convolutional layers in neural networks Jiao et al. (2020), that extract translation-invariant fault patterns in a signal, can be viewed as the addition of domain knowledge into data-driven methods.

In this paper, we propose that unsupervised latent variable models have the potential to facilitate the incorporation of domain knowledge into a fault detection problem. To demonstrate this idea, the loss function of an autoencoder (AE) model is regularized such that, when applied to unseen faulty data, the AE healthy latent representation deviates in a known latent failure direction corresponding to a given fault mode. Ultimately this designed latent space behaviour is then useful for constructing sensitive and interpretable machine health indicators.

In previous work, specialized loss functions have been used to manipulate latent representations for improving supervised classification tasks Li et al. (2018), latent representations have been visualized for different fault modes Booyse et al. (2020), and have been successfully used for constructing informative machine health indicators Balshaw et al. (2022). In other works, augmented training data has been used for data-efficient bearing diagnostics Yu et al. (2021). However, the opportunity of using augmented data, derived from domain knowledge, to shape the latent feature space of an AE with the goal of creating informative and interpretable health indicators has not been widely studied. Therefore, this work intends on making the following contributions.

1.2. Contributions

- Augmented data, as derived from healthy data through a modification of the healthy data towards its expected faulty behaviour, is used as supplementary data for training an AE.
- An AE model is regularized to incorporate domain knowledge into health indicators based on changes in the model's representation with increasing fault severity.
- An interpretable latent representation with diagnostics information is created by including domain knowledge conveyed through augmented data.
- A framework is provided for dealing with the discrepancy between real failure data and a model of the expected failure behaviour.
- The method is applied to a simulated, and experimental dataset.

The remainder of the paper is structured as follows. Section 2 introduces the proposed method, explaining the data preparation, training and evaluation procedures respectively. Results are then presented for a simulated dataset in Section 3 and for the NASA IMS bearing dataset in Section 4. Finally, conclusions and future work are presented in Section 5.

2. METHOD

In this section, a method for incorporating domain knowledge into an unsupervised latent variable model is introduced. Specifically, domain knowledge about bearing fault frequencies is incorporated into an AE model. The intention is to use domain knowledge informed augmented data, as derived from healthy data, to shape the latent space of the AE. As a result, the latent representation of the model should have desirable properties for extracting informative health indicators.

The methodology can be divided into three main parts (See Figure 1). During data preparation (Section 2.1), raw accelerometer signals are processed and healthy data is augmented towards its expected faulty state. Thereafter, the healthy and the augmented data are used to train an AE (Section 2.2). Regularisation of the AE, enabled by including augmented data in the training procedure, enforces that the healthy latent distribution of the AE should deviate in a specific direction as the severity of the fault increases. Finally, during the evaluation phase (Section 2.3), a health indicator is computed for each of the anticipated fault modes to assess the likelihood of a new sample being healthy or faulty, given a particular fault mode.

2.1. Data preparation

In this work, the Squared Envelope Spectrum (SES) is used as input feature to an auto-encoder since it is simple to modify the healthy envelope spectrum towards an expected damaged

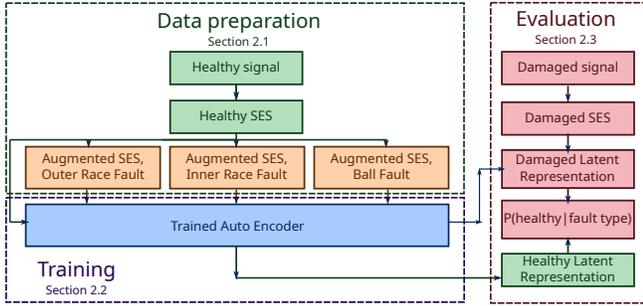


Figure 1. Overview of methodology.

condition. The process of modifying a healthy datapoint towards its expected faulty condition, for a given failure mode, is viewed as data augmentation in this investigation.

Healthy data, $\mathbf{x}_{\text{healthy}}$, is modified by adding a modification signal $\mathbf{x}_{\text{modify}}$ to acquire the augmented data, $\bar{\mathbf{x}}^{(i)}$ for a given fault mode (i).

$$\bar{\mathbf{x}}^{(i)} = \mathbf{x}_{\text{healthy}} + \mathbf{x}_{\text{modify}}^{(i)} \quad (1)$$

Particularly, the amplitude of the healthy squared envelope spectrum at the fault frequency and its first two harmonics corresponding to a given fault mode is increased by $\mathbf{x}_{\text{modify}}$. It should be noted that there are many different ways of achieving this data augmentation, including using the faulty response of a phenomenological model or lumped parameter model or even asking an expert to draw the expected fault behaviour on a graph. In this investigation, simple triangular peaks are added to the expected fault frequency and its harmonics.

A triangular peak x_{peak} as a function of SES frequency f , is defined as:

$$x_{\text{peak}}(f, f_c) = \begin{cases} -|\frac{-2a}{w}(f - f_c)| + a & \text{if } -\frac{w}{2} \leq f - f_c \leq \frac{w}{2} \\ 0 & \text{otherwise} \end{cases}, \quad (2)$$

with a the amplitude of the peak, w the base length of the triangle and f_c the centre frequency for which a high amplitude is associated with a fault.

The total modification signal is obtained by adding the peaks at the fault frequency, f_{fault} , and its harmonics nf_{fault} . The amplitude of the harmonics of the fault frequency decay with frequency and is controlled using the decay parameter α .

$$x_{\text{modify}}(f) = \sum_{n=1}^N e^{-\alpha f_{\text{fault}}(n-1)} x_{\text{peak}}(f, nf_{\text{fault}}) \quad (3)$$

For this investigation the number of peaks, N is selected as

$N = 3$.

Figure 2 shows an example of a healthy envelope spectrum that was modified by adding triangular peaks at frequencies that are expected to correspond to an outer race fault. The healthy signal and the augmented signal are identical, apart from the sections where peaks were added at the fault frequencies.

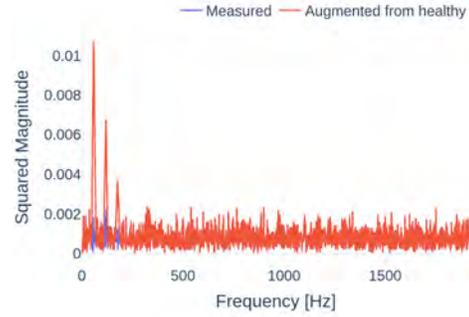


Figure 2. Example of healthy envelope spectrum augmented by adding peaks at expected fault frequencies for an outer race fault.

All healthy training data samples are augmented to additionally obtain an augmented sample for each anticipated fault mode. Both the healthy data and the augmented data are min-max normalized before training.

2.2. Training the auto encoder with specialised loss function

The training phase is applicable during the lifetime of a machine where the bearings are new, have been run in and are assumed to be in a healthy condition. During training, an auto-encoder is used to learn informative latent features from input data. Additionally, the latent space of the auto-encoder is regularised during training such that the augmented data is distributed in the latent space in a way that is beneficial for fault detection.

Specifically, two regularisation terms are used in the loss function in addition to the typical AE reconstruction loss. A first regularisation term enforces that the direction of deviation for the latent healthy data to the augmented data should be maximally different for different fault modes. This ensures that even if the augmentation of the data towards a failed state is not completely representative of reality, the movement of the latent features away from the healthy latent distribution will not be confused with that of another fault mode. This also leads to benefits when computing the projection of the latent representation of a new sample onto an expected failure direction as discussed in Section 2.3. A second regularisation term enforces that the distance from the healthy data cluster in the latent space to the respective augmented clusters should

be similar for different fault modes. This regularisation ensures that the latent space is not disproportionately scaled for a specific fault mode, and simultaneously encodes rudimentary fault severity information into the latent features of the AE since the latent representation of a faulty sample can be compared to that of the augmented samples.

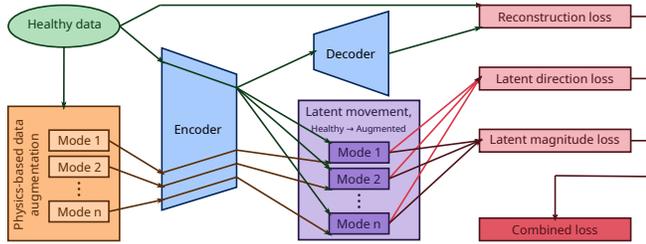


Figure 3. Training Methodology: During forward propagation of the autoencoder, both healthy data and augmented data are separately fed through the encoder. A latent direction loss and latent magnitude loss regularize the latent features whilst a reconstruction loss acts on the decoder output.

Figure 3 shows a diagram explaining the training procedure. During the forward propagation step of training the AE, both the healthy data, and the augmented data for each respective fault mode, are fed through the network separately. The healthy data is fed through both the encoder and the decoder, whilst the augmented data for each respective mode are fed through the encoder only. This is since the latent representation of the augmented data is used only to constrain the latent representation, whilst the reconstructed healthy data is additionally used for computing the conventional AE reconstruction error. After forward propagation, the loss is computed for subsequent backpropagation and the update of the model weights.

The loss function used during training is now described for a randomly selected healthy training example \mathbf{x} and a randomly selected augmented sample $\bar{\mathbf{x}}^{(i)}$, associated with an expected fault mode (i). The combined loss, \mathcal{L} is written as the sum of a reconstruction loss $\mathcal{L}_{\text{reconstruct}}$, acting on the output of the decoder, and the latent feature loss $\mathcal{L}_{\text{latent}}$ acting on the latent representation of the healthy and augmented data.

$$\mathcal{L}(\mathbf{x}) = \mathcal{L}_{\text{reconstruct}}(\mathbf{x}) + \mathcal{L}_{\text{latent}}(\mathbf{x}, \bar{\mathbf{x}}^{(i)}) \quad (4)$$

The mean squared error loss is used as the reconstruction loss as is common in conventional AEs. The purpose of the reconstruction loss is to enforce that low dimensional, informative features are learnt in the latent representation such that the original input can be reconstructed from the latent representation. The reconstruction error is written as:

$$\mathcal{L}_{\text{reconstruct}}(\mathbf{x}) = (\mathbf{x} - g(h(\mathbf{x})))^2, \quad (5)$$

where h and g represent the encoder and decoder of the AE respectively.

The combined loss in Equation 4 further includes a regularisation loss $\mathcal{L}_{\text{latent}}$, that acts on the latent features. This latent loss function consists of two parts, namely $\mathcal{L}_{\text{direction}}$ and $\mathcal{L}_{\text{magnitude}}$.

$$\mathcal{L}_{\text{latent}}(\mathbf{x}) = \lambda_1 \mathcal{L}_{\text{direction}}(\mathbf{x}) + \lambda_2 \mathcal{L}_{\text{magnitude}}(\mathbf{x}) \quad (6)$$

The direction loss enforces that the direction in which a healthy latent cluster is expected to move towards the augmented latent clusters should be maximally different for different fault modes. The magnitude loss ensures that the latent representation is not skewed with respect to a specific latent fault mode. The regularisation hyperparameters, λ_1 and λ_2 scale the importance of the respective loss terms and can be selected based on the loss terms calculated from a validation set.

The magnitude and direction losses are now defined. To simplify the definition of these terms, we introduce $\delta^{(i)}(\mathbf{x})$; the difference between the latent encoding of a healthy sample and the latent encoding of an augmented sample for a given mode (i). Furthermore, let $\mathbf{z} = h(\mathbf{x})$ represent the latent representation of data fed through the encoder of the AE.

$$\begin{aligned} \delta^{(i)}(\mathbf{x}) &= h(\bar{\mathbf{x}}^{(i)}) - h(\mathbf{x}) \\ &= \bar{\mathbf{z}}^{(i)} - \mathbf{z} \end{aligned} \quad (7)$$

The latent movement direction loss is driven by the dot product between the unit vectors of $\delta^{(i)}(\mathbf{x})$ for each fault mode (i). By minimizing the dot product between two vectors, we enforce that the unit vectors of $\delta^{(i)}(\mathbf{x})$ and $\delta^{(j)}(\mathbf{x})$ for fault modes (i) and (j) are pointing in opposite directions. In this work, the unit vectors of $\delta^{(i)}(\mathbf{x})$ are referred to as expected fault directions. The direction loss acting on the latent representation is written as:

$$\mathcal{L}_{\text{direction}}(\mathbf{x}) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{\delta^{(i)}(\mathbf{x})}{\|\delta^{(i)}(\mathbf{x})\|} \cdot \frac{\delta^{(j)}(\mathbf{x})}{\|\delta^{(j)}(\mathbf{x})\|}, \quad (8)$$

adding the unit vector dot products between fault modes for all n expected fault modes.

Finally, the latent movement magnitude loss enforces that the latent augmentation clusters for each mode are equally far from the healthy latent distribution. This ensures that the latent representation is not more sensitive to movement in one

failure direction as compared to the others. Additionally, this loss term ensures that the latent representation does not collapse to a single point with healthy and augmented data in the same location. The latent movement magnitude loss is given as

$$\mathcal{L}_{magnitude}(\mathbf{x}) = \sum_{i=1}^n \left(\|\delta^{(i)}(\mathbf{x})\| - 1 \right)^2. \quad (9)$$

For each batch in the training dataset, the combined loss can be computed and the weights of the autoencoder can be updated by an optimisation algorithm relying on gradients from backpropagation.

2.3. Evaluation of new samples

After the network has been trained, health indicators can be computed for new samples to assess if a fault is present in the bearing and to determine the fault mode by which the bearing is likely failing. A diagram of the evaluation method is shown in Figure 4.

The goal of the evaluation procedure is to evaluate the likelihood that a new sample is still healthy for a given fault mode. To do this, the latent representation of a new sample is projected onto one of the expected failure directions in the latent space. The likelihood of the projected sample can then be estimated based on the distribution of the healthy latent representation as projected onto the same failure direction. Although the direction loss term (Equation 8) enforced that fault directions should be maximally different during training, these fault directions are not explicitly known after optimisation and should be computed.

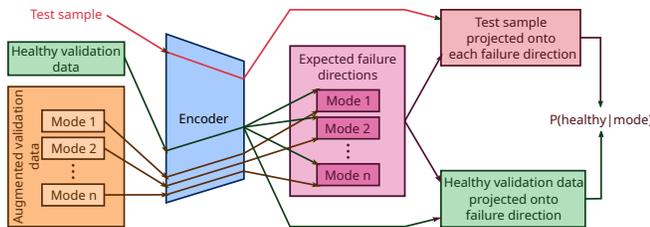


Figure 4. Evaluation method: After computing the expected failure directions in the latent feature space, a new sample is projected onto each of the failure directions so that its likelihood of having failed by a given failure mode can be evaluated.

This is done by computing the expected fault directions from the latent representation of a validation set. The fault directions $\mathbf{v}^{(i)}$ are computed as the normalized unit vector that passes through both the median of the healthy validation data latent representation and the median of the augmented data latent representation for a given failure mode. The latent representation of a new sample from a bearing failing by a given failure mode is expected to deviate from the healthy distribu-

tion in a similar direction than the expected failure direction $\mathbf{v}^{(i)}$. This is since the directions in the latent space were designed to correspond to a given failure through the inclusion of augmented data during the training procedure.

The fault directions $\mathbf{v}^{(i)}$ is given as

$$\mathbf{v}^{(i)} = \text{med} \left\{ \frac{\delta^{(i)}(\mathbf{x})}{\|\delta^{(i)}(\mathbf{x})\|} \text{ for all } \mathbf{x} \text{ in validation set} \right\}. \quad (10)$$

With the expected fault directions $\mathbf{v}^{(i)}$ calculated, the scalar projection of the latent representation in an expected fault direction $\mathbf{v}^{(i)}$ can be computed. This is done by taking the dot product between the latent representation of a sample \mathbf{z} and an expected failure direction $\mathbf{v}^{(i)}$.

$$z_{proj}^{(i)} = \mathbf{z} \cdot \mathbf{v}^{(i)} \quad (11)$$

This projection is demonstrated on the left hand side of Figure 5.

Finally, the likelihood of a sample being healthy is calculated for each failure direction (i). To do this, the distribution of the projection of the healthy data onto a certain failure direction (i), is assumed to be Gaussian with mean $\mu^{(i)}$ and standard deviation $\sigma^{(i)}$ for positive values of z_{proj} . This ensures that the likelihood of a new sample can be computed based on the healthy distribution parameters $\mu^{(i)}$ and $\sigma^{(i)}$ in the fault direction (i). Furthermore, the likelihood of the sample is assumed to follow a uniform distribution for negative values of z_{proj} , since a deviation of the latent features in the opposite direction of a failure direction is not expected to correspond to a fault. The expression for evaluating the likelihood of a new sample in failure direction (i) is shown in equation Eq. 12.

$$p(z_{proj}^{(i)} | \mu^{(i)}, \sigma^{(i)}) = \begin{cases} \frac{1}{\sigma^{(i)}\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{z_{proj}^{(i)} - \mu^{(i)}}{\sigma^{(i)}}\right)^2\right) & z_{proj} > 0 \\ \frac{1}{\sigma^{(i)}\sqrt{2\pi}} & z_{proj} \leq 0 \end{cases} \quad (12)$$

The evaluation of the likelihood of new samples in the projected dimension is demonstrated on the right hand side of Figure 5. A sample that is considered likely for a given failure direction, can be highly unlikely for a different failure direction.

By evaluating the likelihood of a sample for a given fault mode (projection onto a fault direction) a fault-mode-specific health indicator can be obtained from the latent representation movement with increasing fault severity. The condition that fault directions should be maximally different, as enforced during training, ensures that faulty data from different failure modes are not confused. The projection of new samples

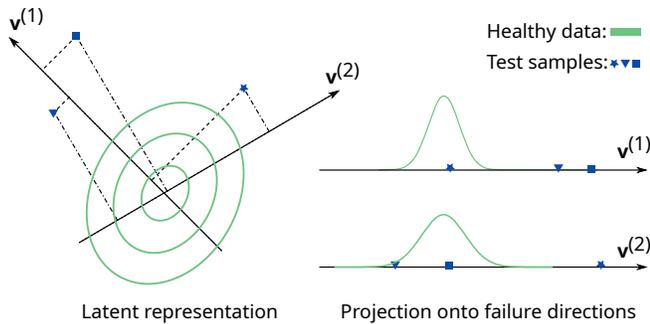


Figure 5. Evaluation of a new test sample: The likelihood of a new sample for a given failure direction $v^{(i)}$ is calculated based on the projection of the healthy data onto the failure directions in the latent representation.

onto the failure directions further ensures that the discrepancy between the actual fault behaviour and the expected fault behaviour, as communicated to the model by the use of augmented data, is less relevant.

The methodology is now demonstrated on two datasets.

3. PHENOMENOLOGICAL BEARING DATASET

In this section, a dataset generated from a phenomenological model based on that of McFadden & Smith (1984) is used to demonstrate the proposed fault detection method. The model used is further extended to include random variations of the amplitude of the transient excitations and random ball slip. The benefit of using a phenomenological model to demonstrate the proposed method is that all potential fault modes can be simulated. This means that the same model, trained on healthy data, can be evaluated on each of the expected degradation paths. The specifications of the phenomenological dataset are listed in Table 1.

The SES for the healthy data is computed from the time domain signal, whereafter the augmented data is computed from the healthy SES as described in the methodology. Figure 6 shows an example of the squared envelope spectrum at the highest fault severity of the data as compared to the augmented data. The figure demonstrates that the augmentation of the data can be imperfect and does not need to be completely representative of reality to improve the separability and interpretability of the latent representation.

With the healthy and augmented data available, the AE model can be trained. The specifications of the AE models and data augmentation used for each dataset in this investigation are shown in Table 2. In this example, the latent feature representation dimensionality (AE bottleneck size) is chosen as two, so that the latent features can be easily visualized in two-dimensional space. It should be mentioned that the latent representation dimensionality can be viewed as a hyperparameter that needs to be chosen similar to any other AE hyperpa-

Table 1. Specifications for phenomenological dataset.

Model properties	
Transient peak range for different severities	0-1 m/s^2
Modulation amplitude for inner race fault	1
Variance of slip	0.001 rad
Measurement noise standard deviation	0.2 m/s^2
Transient amplitude standard deviation	0.05 m/s^2
Ball diameter	8.4 mm
Pitch circle diameter	71.5 mm
Number of balls	16
Contact angle	15.7 deg
SDOF stiffness	2×10^{13} N/m
SDOF damping ratio	0.05
SDOF natural frequency	4230 Hz
Constant rotation speed	500 RPM
Sampling frequency	38400 Hz
Signal duration	1 s
Dataset Properties	
Healthy training samples	450
Augmented training samples per fault mode	450
Failure modes considered	3
Healthy validation samples	50
Augmented validation samples per fault mode	50
Damaged test samples	500

Table 2. Specifications AE model and data augmentation used for each dataset.

Specifications	Phenomenological Dataset	IMS Dataset
Model specifications		
Input size	1920	1024
Encoder layer 1	754	402
Bottleneck layer	2	2
Decoder layer 1	754	402
Output size	1920	1024
Activation central layers	ReLU	ReLU
Activation output layer	Tanh	Tanh
Direction regularisation, λ_1	1×10^{-2}	1×10^{-1}
Magnitude regularisation, λ_2	1×10^{-2}	1×10^{-1}
Augmentation specifications		
Peak amplitude a	5×10^{-3}	5×10^{-3}
Decay parameter, α	2×10^{-2}	2×10^{-2}
Base width w	20Hz	20Hz

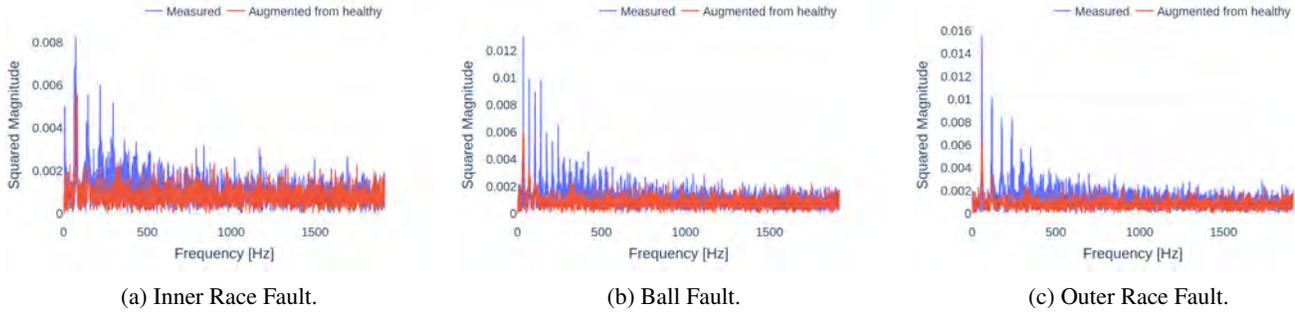


Figure 6. Phenomenological dataset: Comparison of augmented data squared envelope spectrum with failure data at maximum severity. Although there are significant discrepancies between the expected and true fault behaviour, the augmented data is still useful for constraining the latent representation of the autoencoder model

parameter. There is no requirement for a relationship between the dimensionality of the latent space and the number of fault modes that the model accounts for.

Figure 7 shows the latent representation of data after the model has been trained. The latent representation of the faulty test data is shown in Figure 7a together with augmented data from the validation set. The expected failure directions that pass through the median of the healthy and augmented data for each mode are also shown as straight lines. The failure directions are separated in the latent space and the augmented clusters are equally far away from the healthy data cluster, demonstrating that the direction and magnitude loss in Equation 6 was successful in constraining the latent representation. For a given fault mode, the initially healthy latent distribution moves in a direction in the latent representation with increasing fault severity that is consistent with the expected fault direction. For instance, if an outer race fault is present, the latent distribution will move in the general direction indicated by the expected fault direction of the outer race fault.

Therefore, including domain knowledge through augmented data lead to a separated and interpretable latent representation for the AE. As a result, this ensures that informative health indicators can be computed from the latent representation. Additionally, this makes the latent representation interpretable, since anomalous samples not related to bearing faults will likely be distributed in the latent representation in a way that is not consistent with what is expected from the encoded bearing fault behaviour.

In the next step of the methodology, the samples are projected onto the failure directions as visualised in Figure 7 and the likelihood of a test sample is evaluated from Equation 12.

Figure 8 shows the negative log-likelihood for each of the expected failure directions. Each sub-figure shows the result for a certain ground truth fault mode. For a given fault mode, the negative log-likelihood health indicator rises sharply for faulty data projected onto the failure direction that corresponds

to the ground truth fault mode. For example, in Figure 8a the ground truth fault mode is a ball fault. Consequently, the negative log-likelihood of the data projected onto the ball fault direction increases with increasing fault severity. In contrast, the negative log-likelihood of data projected onto fault directions that are not associated with a ball fault remains comparatively low. Thereby, an informative healthy indicator is obtained that can indicate faulty behaviour and simultaneously provide diagnostics information about the fault mode.

In the next section, a similar analysis is conducted on the IMS dataset.

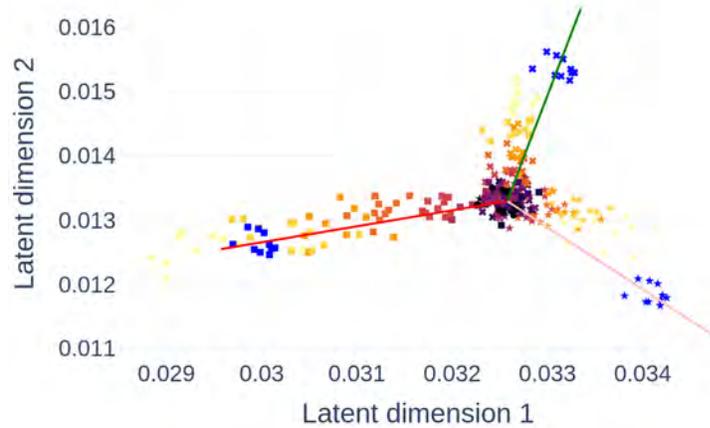
4. APPLICATION ON NASA IMS EXPERIMENTAL DATASET

The NASA IMS dataset (Qiu et al., 2006) is a popular dataset used in bearing condition monitoring. It consists of three separate run-to-failure tests, each including data for four bearings. Ground truth labels of the failure modes in which a bearing had failed (Inner Race, Outer Race or Ball Fault) are available for four of the 12 bearings. This investigation will focus on the four datasets with labelled ground truth labels in order to check if the proposed method is successful in detecting a fault associated with a particular fault mode. Information about the datasets that are used in this investigation is shown in Table 3.

The healthy records for training are chosen in accordance with that of Liu & Gryllias (2020) with some run-in records not used for training. The remaining records are used as the test set during the evaluation phase. Table 3 shows the record numbers used for training as well as the ground truth label for each of the datasets considered.

To illustrate the data augmentation process, Figure 9 shows an example of test samples from the IMS dataset at a high severity compared to the augmented data for the same fault mode.

The SES seems to be an effective way of extracting fault information from the two outer race fault datasets, but is less



(a) Test samples (shaded), augmented validation samples (blue), healthy validation samples (black) and expected failure directions (colored lines).



(b) Legend.

Figure 7. Latent representation after training. Shaded data points represent test data at a certain fault severity. Shapes represent different fault modes

Table 3. IMS dataset information.

Dataset	Channel	Recorded Failure Mode	Healthy record numbers	Training Samples	Validation Samples
1	Bearing 3, Channel 5	Inner Race	200-600	360	40
1	Bearing 4, Channel 7	Ball	200-600	360	40
2	Bearing 1, Channel 1	Outer Race	50-300	225	25
3	Bearing 3, Channel 3	Outer Race	50-300	225	25

effective for inner race and ball fault datasets, leading to large discrepancies between the true and augmented data. As a result, the latent representation for the fault data shown in Figure 10 is not well structured in the latent representation for all of the datasets. However, for the outer race fault of test 2, bearing 1 the latent features clearly move along the outer race failure direction with increasing fault severity.

The negative log-likelihood of a sample projected onto a given failure direction is shown in Figure 11. The method appears to be effective for the outer race and ball fault datasets, with the negative log-likelihood increasing for the ground truth failure direction. However, the negative log associated with

the inner race fault does not seem to be more sensitive to the inner race fault as compared to the negative log-likelihood associated with the other fault modes. This is due to the latent space being uninformative after the completion of training, since the envelope spectrum that is used as input is not sensitive to the inner race fault.

The effectiveness of the method is reliant on a correspondence between the augmented data and the true failure behaviour so that the latent representation can be sensitive to a given failure mode. Consequently, it is expected that the effectiveness of the method is dependant on how informative the input features are. In this analysis, where a simple input feature such as the envelope spectrum was used without any additional pre-processing such as band pass filtering around informative frequency bands, it is clear that the envelope spectrum is not sensitive to certain fault types, and as a result the latent feature representations were not informative for these failure modes. In future work this limitation could be addressed by training models on the time domain data directly, or by using more advanced features from signal processing.

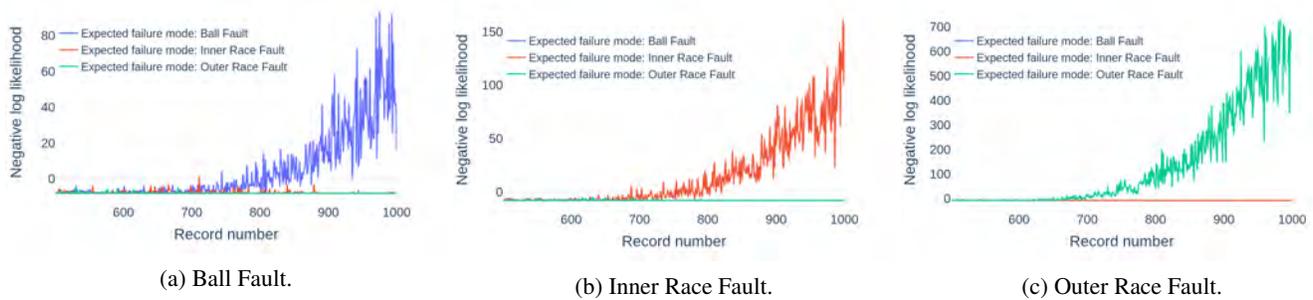


Figure 8. Phenomenological dataset: Negative log-likelihood with increasing fault severity. Sub figures show data for different ground truth fault modes. Each trace on a sub-figure shows the log-likelihood of the test data for an expected failure mode.

5. CONCLUSION AND FUTURE WORK

This paper presents a new way of encoding domain knowledge into the latent representation of an AE by using augmented data. Results on a phenomenological dataset demonstrate that incorporation of domain knowledge leads to an interpretable latent representation that is useful for constructing informative health indicators for fault detection and diagnostics. Furthermore, the method is applied to the experimental NASA IMS dataset. The method proves to be effective for three of the four IMS datasets considered, with the success of the method being reliant on how sensitive the input features are to damage.

In future work, the proposed method can be extended to act on time-frequency maps or even directly on time series data, where the data augmentation can be facilitated by a phenomenological model. This can ensure that hidden fault information is not withheld from the model whilst still allowing for the incorporation of domain knowledge. Furthermore, the sensitivity of the method to inaccuracies in modelling the expected fault behaviour, the chosen size of the latent representation, training batch sizes and the model architecture needs to be determined.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of the European Commission under the Marie Skłodowska-Curie program through the ETN MOIRA project (GA 955681) and the support of the Flemish Government under the “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen” programme.

REFERENCES

Balshaw, R., Heyns, P. S., Wilke, D. N., & Schmidt, S. (2022, April). Importance of temporal preserving latent analysis for latent variable models in fault diagnostics of rotating machinery. *Mechanical Systems and Signal Processing*,

168, 108663. doi: 10.1016/j.ymsp.2021.108663

Booyse, W., Wilke, D. N., & Heyns, S. (2020, June). Deep digital twins for detection, diagnostics and prognostics. *Mechanical Systems and Signal Processing*, 140, 106612. doi: 10.1016/j.ymsp.2019.106612

Cao, H., Niu, L., Xi, S., & Chen, X. (2018, March). Mechanical model development of rolling bearing-rotor systems: A review. *Mechanical Systems and Signal Processing*, 102, 37–58. doi: 10.1016/j.ymsp.2017.09.023

Cerrada, M., Sánchez, R.-V., Li, C., Pacheco, F., Cabrera, D., Valente de Oliveira, J., & Vásquez, R. E. (2018, January). A review on data-driven fault severity assessment in rolling bearings. *Mechanical Systems and Signal Processing*, 99, 169–196. doi: 10.1016/j.ymsp.2017.06.012

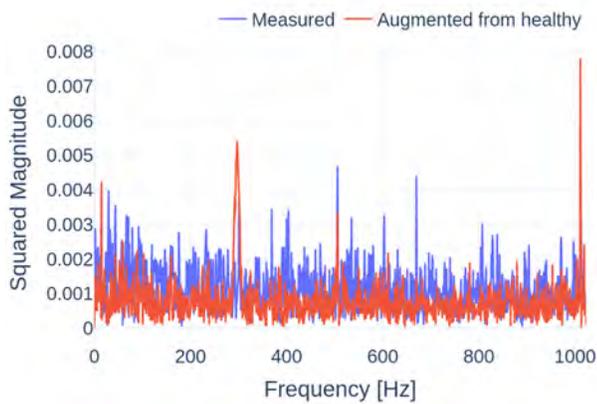
Hoang, D.-T., & Kang, H.-J. (2019, March). A survey on Deep Learning based bearing fault diagnosis. *Neurocomputing*, 335, 327–335. doi: 10.1016/j.neucom.2018.06.078

Jiao, J., Zhao, M., Lin, J., & Liang, K. (2020, December). A comprehensive review on convolutional neural network in machine fault diagnosis. *Neurocomputing*, 417, 36–63. doi: 10.1016/j.neucom.2020.07.088

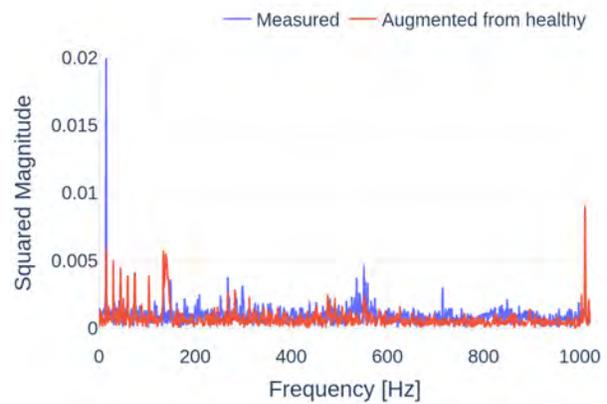
Lee, J., Wu, F., Zhao, W., Ghaffari, M., & Liao, L. (2014). Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications. *Mechanical Systems and Signal Processing*, 21. doi: <http://dx.doi.org/10.1016/j.ymsp.2013.06.004>

Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2018, May). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, 104, 799–834. doi: 10.1016/j.ymsp.2017.11.016

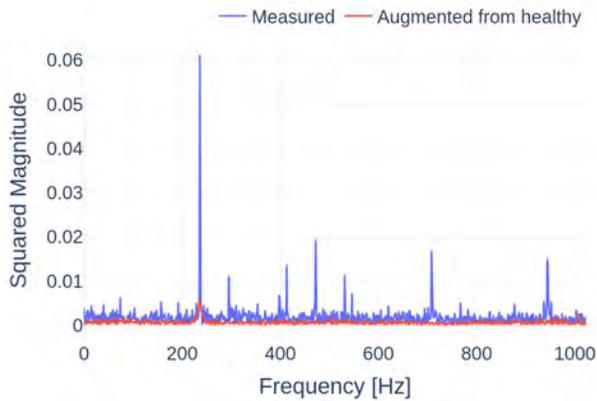
Leturiondo, U., Salgado, O., Ciani, L., Galar, D., & Catealani, M. (2017, October). Architecture for hybrid modelling and its application to diagnosis and prognosis with missing data. *Measurement*, 108, 152–162. doi: 10.1016/j.measurement.2017.02.003



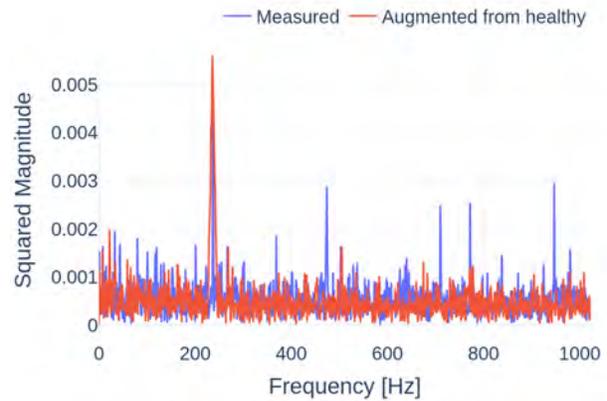
(a) IMS test 1, Bearing 1: Inner Race Fault.



(b) IMS test 1, Bearing 4: Ball Fault.



(c) IMS test 2, Bearing 1: Outer Race Fault.



(d) IMS test 3, Bearing 3: Outer Race Fault.

Figure 9. Squared envelope spectrum for augmented data and data at high severity.

Li, X., Zhang, W., & Ding, Q. (2018, October). A robust intelligent fault diagnosis method for rolling element bearings based on deep distance metric learning. *Neurocomputing*, 310, 77–95. doi: 10.1016/j.neucom.2018.05.021

Liao, L., & Kottig, F. (2014, March). Review of Hybrid Prognostics Approaches for Remaining Useful Life Prediction of Engineered Systems, and an Application to Battery Life Prediction. *IEEE Transactions on Reliability*, 63(1), 191–207. doi: 10.1109/TR.2014.2299152

Liu, C., & Gryllias, K. (2020, June). A semi-supervised Support Vector Data Description-based fault detection method for rolling element bearings based on cyclic spectral analysis. *Mechanical Systems and Signal Processing*, 140, 106682. doi: 10.1016/j.ymssp.2020.106682

Liu, C., Mauricio, A., Qi, J., Peng, D., & Gryllias, K. (2020, November). Domain Adaptation Digital Twin for Rolling Element Bearing Prognostics. *Annual Confer-*

ence of the PHM Society, 12(1), 10. doi: 10.36001/phm-conf.2020.v12i1.1294

McFadden, P., & Smith, J. (1984, September). Model for the vibration produced by a single point defect in a rolling element bearing. *Journal of Sound and Vibration*, 96(1), 69–82. doi: 10.1016/0022-460X(84)90595-9

Qiu, H., Lee, J., Lin, J., & Yu, G. (2006, February). Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. *Journal of Sound and Vibration*, 289(4-5), 1066–1090. doi: 10.1016/j.jsv.2005.03.007

Randall, R. B., & Antoni, J. (2011, February). Rolling element bearing diagnostics—A tutorial. *Mechanical Systems and Signal Processing*, 25(2), 485–520. doi: 10.1016/j.ymssp.2010.07.017

Sadoughi, M., & Hu, C. (2019, June). Physics-Based Convolutional Neural Network for Fault Diagnosis of Rolling

Element Bearings. *IEEE Sensors Journal*, 19(11), 4181–4192. doi: 10.1109/JSEN.2019.2898634

Shen, S., Lu, H., Sadoughi, M., Hu, C., Nemani, V., Thelen, A., . . . Kenny, S. (2021, August). A physics-informed deep learning approach for bearing fault detection. *Engineering Applications of Artificial Intelligence*, 103, 104295. doi: 10.1016/j.engappai.2021.104295

Yu, K., Lin, T. R., Ma, H., Li, X., & Li, X. (2021, January). A multi-stage semi-supervised learning approach for intelligent fault diagnosis of rolling bearing using data augmentation and metric learning. *Mechanical Systems and Signal Processing*, 146, 107043. doi: 10.1016/j.ymssp.2020.107043

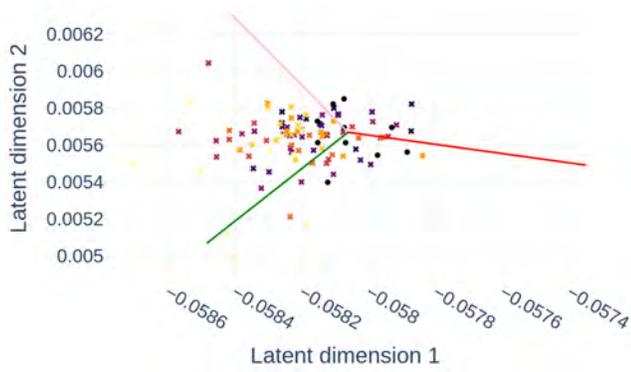
BIOGRAPHIES



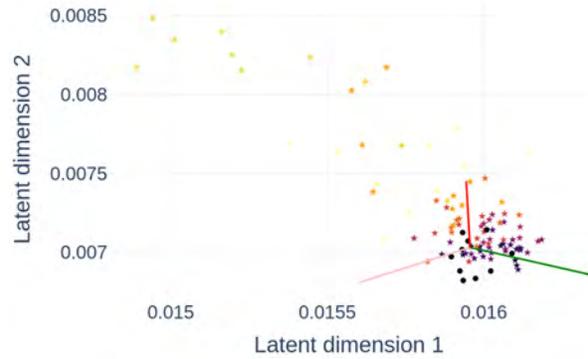
Douw Marx received his B.S. and M.Sc degree in mechanical engineering from the University of Pretoria, South-Africa. He joined the Noise and Vibration Research Group in the Department of Mechanical Engineering, KU Leuven, Belgium as a PhD researcher in 2021. His research interests include hybrid approaches for fault detection, signal processing and deep learning.



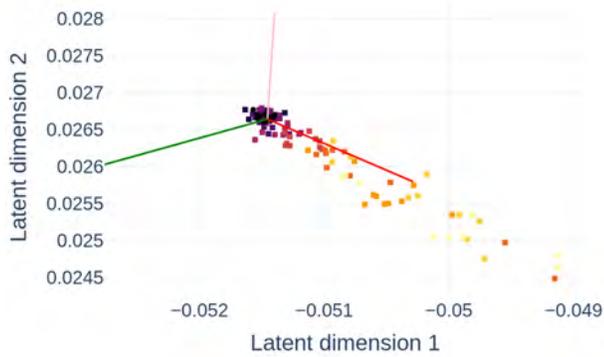
Konstantinos Gryllias holds a 5 years engineering diploma degree and a PhD degree in Mechanical Engineering from National Technical University of Athens, Greece. He holds an associate professor position on vibroacoustics of machines and transportation systems at the Department of Mechanical Engineering of KU Leuven, Belgium. He is also the manager of the University Core Lab Dynamics in Mechanical & Mechatronic Systems DMMS-M of Flanders Make, Belgium. His research interests lie in the fields of condition monitoring, signal processing, prognostics and health management of mech. & mechatronic systems.



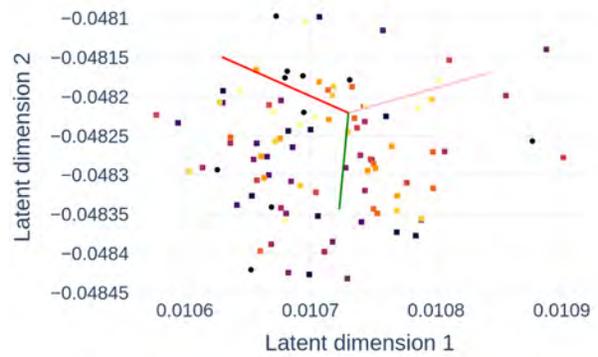
(a) IMS test 1, Bearing 1: Inner Race.



(b) IMS test 1, Bearing 4: Ball.



(c) IMS test 2, Bearing 1: Outer Race.

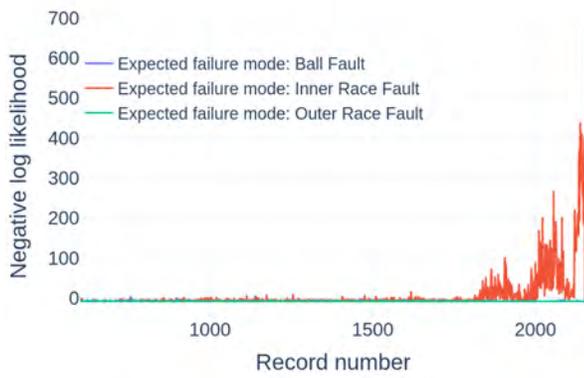


(d) IMS test 3, Bearing 3: Outer Race.

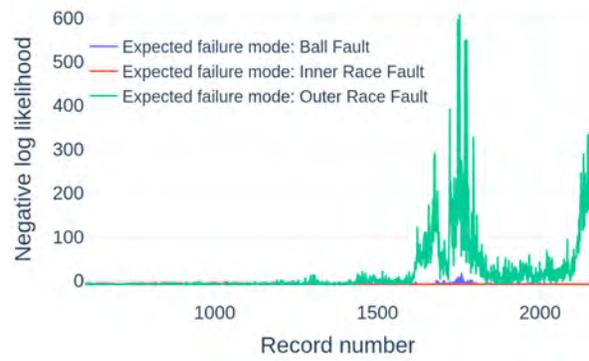


(e) Legend.

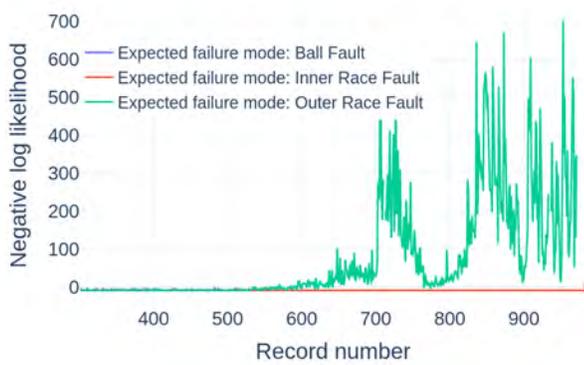
Figure 10. IMS dataset latent representation at different severities.



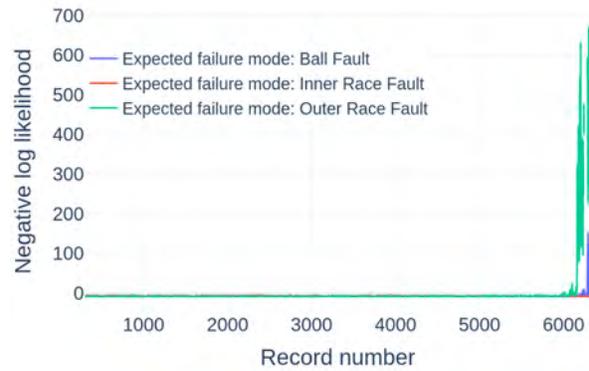
(a) IMS test 1, Bearing 1: Inner Race Fault.



(b) IMS test 1, Bearing 4: Ball Fault.



(c) IMS test 2, Bearing 1: Outer Race Fault.



(d) IMS test 3, Bearing 3: Outer Race Fault.

Figure 11. IMS dataset: Negative log likelihood of measurements onto projected onto respective fault modes

Estimation of Wind Turbine Performance Degradation with Deep Neural Networks

Manuel S. Mathew¹, Surya Teja Kandukuri², Christian W. Omlin³

^{1,2,3}*University of Agder, Jon Lilletuns Vei 9, 4879 Grimstad, Norway*

manuel.s.mathew@uia.no

surya.kandukuri@uia.no

christian.omlin@uia.no

ABSTRACT

In this paper, we estimate the age-related performance degradation of a wind turbine working under Norwegian environment, based on a deep neural network model. Ten years of high-resolution operational data from a 2 MW wind turbine were used for the analysis. Operational data of the turbine, between cut-in and rated wind velocities, were considered, which were pre-processed to eliminate outliers and noises. Based on the SHapley Additive exPlanations of a preliminary performance model, a benchmark performance model for the turbine was developed with deep neural networks. An efficiency index is proposed to gauge the age-related performance degradation of the turbine, which compares measured performances of the turbine over the years with corresponding bench marked performance. On an average, the efficiency index of the turbine is found to decline by 0.64 percent annually, which is comparable with the degradation patterns reported under similar studies from the UK and the US.

1. INTRODUCTION

With the current emphasis on clean and secure energy supply, the global wind power sector is growing significantly in the recent years. From the cumulative wind power installations of 743 GW in 2020 (Global Wind Energy Council, 2021), the global wind capacity may reach up to 6044 GW by 2050 (International Renewable Energy Agency, 2019). This could make wind to be one of the major energy resources, contributing more than one-third of the total global energy demand. Hence, wind energy would play a significant role in the future clean energy scenarios.

Power generated from the turbines over its life span is one of the most important factors deciding the technical feasibility and economic viability of any wind energy project. The normal life span of a turbine could vary from 20 to 25 years,

depending on the design features and operational environments (Adedipe, & Shafiee, 2021; Ziegler, Gonzalez, Rubert, Smolka, & Melero, 2018). Some of the recent studies have reported longer life span for different wind energy projects, ranging from 25 to 40 years and averaged to 29.6 years (Wiser, & Bolinger, 2019). Performance of the wind turbines decline gradually during this life period. There are several reasons for this age-related performance degradation. The mechanical wear and tear of various components over time could affect the turbine's performance, reliability, as well as efficiency (Hamilton, Millstein, Bolinger, Wiser, & Jeong, 2020; Pan, Hong, Chen, Feng, & Wu, 2021). Another major reason for this declining performance is the reduction in aerodynamic efficiency due to material erosion over the blade tips (Sareen, Sapre, & Selig, 2014; Zweiri et al., 2022).

Despite its significant influence on the techno-economic viability of wind energy projects, the age-related efficiency reductions in these systems are not systematically analyzed extensively. As reported by Staffell and Green (2014), most of the earlier studies focused on the component level system reliability and availability.

One of the earliest studies addressing the age-related efficiency degradation was by Hughes (2012), where ten years of operational data from windfarms in the UK and Denmark were used. Performance degradation was represented as the variations in normalized load factor (capacity factor) over the years. The normalized load factor was computed based on monthly productions from the windfarms and the corresponding indices on average wind speeds. The load factor for the UK onshore windfarms declined by 13% within 15 years of its life, whereas the corresponding decline reported for Danish turbines was 4%. Offshore farms showed faster degradation compared to the onshore systems. Similar results were also reported by Staffell and Green (2014) in which 282 windfarms in the UK were considered. The ideal power curve of the turbines in

Manuel S Mathew et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

windfarms, along with the corresponding wind speed simulated using the NASA model, were used to compute the theoretical power production. This is then compared with the reported monthly load factors to estimate the performance decay. Under this study, turbines were found to lose $1.6 \pm 0.2\%$ of their output per year.

In later years, studies on Swedish wind turbines were conducted by Olauson, Edström and Rydén (2017). They found that the turbines could suffer 6 % reduction in capacity factor during their lifetime. Similar performance reduction is reported in the case of German turbines as well (Germer, & Kleidon, 2019). In a more comprehensive analysis involving 917 plants, Hamilton et al. (2020) quantified the age related performance changes in the US windfarms, interestingly, in a policy perspective as well. Under this study, the capacity factor point percentages were reported to decline by 0.53% and 0.17% for older and newer projects respectively.

The above studies have significantly contributed in identifying and understanding the age-related performance issues in wind turbines. Different rates of degradation reported in these studies highlights the regional and site-specific nature of the issue. Nevertheless, these studies are based on the cumulative data from several windfarms, collected from public sources. With the site-specific nature of the degradation pattern, an analysis based on the data from a specific farm/turbine could give a better insight in to the issue. With the extensive deployment of SCADA systems in windfarms, time series performance data from the turbines are available which can be used to develop data driven degradation models as suggested by Astolfi, Castellani, Lombardi and Terzi (2021). Compared to the monthly averaged information used in some of the previous studies, analyses based on the high-resolution SCADA data can bring out more precise understanding on the issue.

In view of this, Dai, Yang, Cao, Liu and Long (2018) analyzed the aging pattern of wind turbines based on SCADA data. However, to eliminate the effects of weather conditions and turbine's operational parameters on the results, the power variations above the rated speed were only considered in the analysis. As the effect of power deficits are more prominent from cut-in to rated wind speeds, inclusion of data from this dynamic operating region of the turbine is essential for such analysis. Study by Kim and Kim (2021), based on the SCADA data, uses the turbine's power curve for performance comparison. As discussed in Veena, Mathew and Petra (2020), limitations of manufacturer's power curve in understanding the site specific dynamics of the velocity-power response of the turbines has been well established in several previous studies. Further, the analysis is based on four years performance of the turbines, which may not be sufficient to capture the time series performance degradation. Other significant studies based on SCADA data, focused on same turbine model installed at Irish and Italian sites, can be seen in Byrne, Astolfi, Castellani and Hewitt (2020) and

Astolfi, Byrne and Castellani (2021). The turbine at the Italian site showed an efficiency reduction of 1.5% over 12 years, whereas the corresponding degradation at the Irish site was 8.8%. These studies further reinstate the site dependencies of the performance degradation phenomena.

To the best of the authors' knowledge, studies on the age-related degradation pattern in turbines working under the Norwegian environment have not yet been reported. In this paper, we present such a first-time analysis based on the time series performance of a 2 MW wind turbine installed at a Norwegian site.

One of the distinct features of this study is the deep neural network (DNN) based site specific benchmark model. The strength of neural networks is that it can approximate any Borel measurable function from a finite dimensional space to another, if sufficient amount of hidden neurons are present in the network (Goodfellow, Bengio, & Courville, 2016). Compared to conventional machine learning (ML) algorithms, neural networks perform extremely well in learning non-linear relationships between variables (Somers, & Casal, 2009). DNNs improve on neural networks with the presence of two or more hidden layers between the input layer, and the output layer. Furthermore, traditional ML algorithms require feature engineering i.e., features must be extracted from the data to learn its relationship with the target. However, DNNs extract the necessary features from the data based on the learning task at hand. This allows the model to be trained with raw input data to learn the underlying representations (Janiesch, Zschech, & Heinrich, 2021).

Other features of the study are the improvement and explainability of the DNN models through SHapley Additive exPlanations (SHAP) analysis and the proposed performance deficit index based on the overall efficiency ratio of the turbine.

After this introductory section, the paper is arranged as follows. Initially, the description of the data used for the analysis and its preprocessing methods are discussed. This is followed by the details of a preliminary DNN based performance model and its SHAP analysis. Architecture of the proposed DNN benchmarking model is then introduced along with the model performance analysis based on different error metrics. The efficiency index for quantifying the age-related performance deficit is then defined and the performance degradation of the turbine over the years, estimated based on the efficiency index, is presented.

2. DATA DESCRIPTION AND PREPROCESSING

A pitch-controlled wind turbine with 2 MW rated capacity, installed in a Norwegian site, was chosen for the study. The turbine has cut-in, rated and cut-out wind speeds of 3.5 m/s, 15 m/s, and 25 m/s respectively. The turbine has a rotor diameter of 82.4 m, and the system is installed over a tower

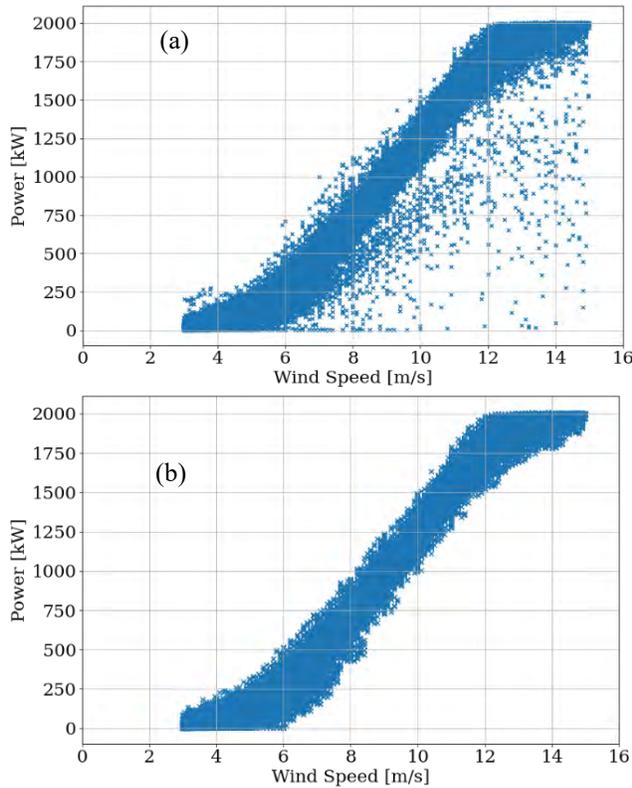


Figure 1. (a) Raw performance data of the turbine between the cut-in and rated wind speeds. (b) data after cleaning the outliers using DBSCAN

Table 1. Details of the SCADA data used for the study

No	Measurements
1	Wind speed
2	Standard deviation of wind speeds
3	Pitch positions of the blades and the pitch reference
4	Yaw position
5	Generator RPM
6	Main shaft RPM
7	Power

of 70 m height. SCADA data from the turbine during the period 2007 to 2017, at 10 min temporal resolution, were collected for the study (Under the non-disclosure agreement, the data cannot be shared with this paper). Various measurements related to the operational and control settings of the turbine were available out of which the most relevant parameters (see Table 1) were considered for the analysis.

Wind turbines have two distinct operational regions viz. the dynamic region corresponding to the cut-in to rated wind

velocities, and the deterministic region corresponding to the rated to cut-out velocities (Veena et al., 2020). Out of these, performance of the turbine between the cut-in and rated wind velocities were considered for this study. This is because the performance degradation of the turbine is expected to be prominently observable between the cut-in to rated velocities. Above the rated velocity, output is regulated to rated power by the control system and hence the degradation would be masked.

The data corresponding to the cut-in to rated region as above contains outliers and noises as shown in Figure 1 (a). These outliers are caused by various reasons like malfunctioning of sensors and logs, downtime of the turbine, power curtailments, weather related factors like icing etc. To eliminate these anomalies, the data has been filtered using density-based spatial clustering of applications with noise (DBSCAN), proposed by Ester, Kriegel, Sander and Xu (1996). DBSCAN is a clustering method which efficiently identifies the arbitrary-shaped clusters and noises in the dataset and thereby filters and cleans the undesired data outliers. Figure 1 (b) shows the performance data from the turbine over the study period, cleaned using DBSCAN.

3. PRELIMINARY MODEL AND SHAP ANALYSIS

Data on various operational parameters were available in the dataset. Out of these, features listed in Table 1 were chosen for the analysis based on the Pearson and Spearman correlations of these variables with the power generated by the turbine.

With these input features, a preliminary performance model for the turbine was developed based on the data from 2007. Purpose of this model was to enhance the model interpretability through explainable AI (XAI) (in contrast to the black box approach in traditional AI methods). The datasets were divided into three groups for training (60%), validation (20%), and testing (20%). The preliminary model was developed using the deep neural network architecture with back propagation of errors for training. Under the error analysis, the preliminary model showed an MAE, RMSE, and MSE of 27.76 and 31.90, and 1017.76 respectively on the test data previously not seen by the model. With this acceptable accuracy, the model was further analyzed using SHAP, introduced by Lundberg and Lee (2017). SHAP, which is a unified approach to interpret the model’s predictions, is used here to explain the prediction of power generated at an instance by calculating the respective contribution from each of the features selected for the model development. SHAP unified seven different methods in explainable AI to provide a framework to interpret the predictions made by a machine learning model, both locally (for a single instance) and globally (across N number of instances). SHAP is based on Shapley values (Shapley, 1952), a collaborative game theory method that involves fairly distributing both gains and costs to actors working in a coalition. The mathematical

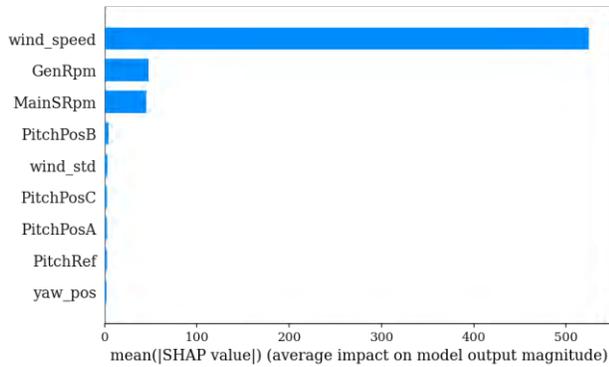


Figure 2. The average SHAP values for various input features for the preliminary model

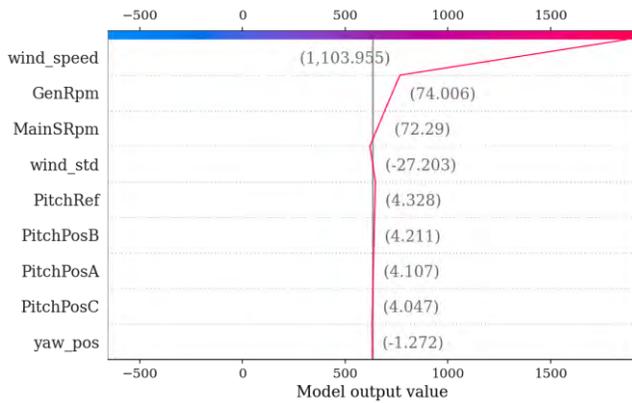


Figure 3. SHAP values for the preliminary model corresponding to a single sample

formulations for computing the SHAP values can be found in Lundberg, and Lee, 2017 as well as Molnar, 2020. For robust calculation of the contributions from each feature and to verify how the model uses these features as a whole for making predictions, 100 random samples were passed on to the model and their corresponding SHAP values has been calculated using the Kernel SHAP method, which is a model-agnostic method and is detailed in Lundberg and Lee (2017). The mean absolutes of the SHAP values, corresponding to each feature, were computed, and are shown in Figure 2. The wind speed has the highest mean absolute value of SHAP, which is then followed by generator and rotor shaft RPMs respectively.

Similar result is observed in the SHAP values in local explainability (in case of a single sample) as well, which is presented in Figure 3. The figure shows how each feature contributes towards pushing the prediction away from the base value. The base value or the expected value is the average of the model outputs over the samples. The red bars indicate an increase in power prediction over this base value corresponding to a particular feature and the blue shows a decrease in prediction. Here also, wind speed and generator

& rotor shaft RPMs form the significant contributions towards the power prediction.

Though the SHAP analysis do not imply causalities explicitly, it helps in interpreting the model’s predictions by explaining the contribution of each feature towards the model output and hence in finding the feature saliency in a model prediction.

4. BENCHMARK MODEL

A site-specific performance model for the turbine, based on its operational data in 2007, was developed for benchmarking the turbine’s performance. This benchmarked performance can be compared with the turbine’s productivity in later years for identifying the age-related performance degradations.

From the SHAP analysis, it is evident that the predictions made by the preliminary model is mostly explained by the wind speed, followed by speeds of the generator and the main shaft. Contributions from other features are relatively low.

Further, some of these measurements were not consistently available during the later years in which the turbine performance is to be compared with the benchmarked performance. In view of these, wind velocity, generator speed (as the speeds of the main shaft and generator are correlated through the gear ratio), and the yaw position were chosen as input features of the proposed benchmark model. These chosen model inputs were further tested for multicollinearity using the variation influence factor (VIF) analysis and found to be within acceptable limits (Sulaiman et al., 2021). The data were then divided into three groups for training (60%), validation (20%) and testing (20%). As in the preliminary model, the benchmark model was also developed using deep neural network with back propagation of errors for training. He-Normal initialization (He, Zhang, Ren, & Sun, 2015) has been done for the kernel and rectified linear unit (ReLU) was used for the activation functions. L2 regularization was used for the kernels and adaptive moment estimation (Adam) optimizer was used for the model development. The model architecture was optimized through iterations taking MAE as the principal error measure while RMSE and MSE were also monitored. The model was trained and validated under 100 epochs and the best performing model across the three monitored errors was chosen. Caution was taken to avoid any over or under fitting of the model by tracking the errors in training and validation during the model development, as shown in Figure 4. As evident from the figures, both training and validation errors are under reasonable limits, which rules out the possibility of underfitting. Further, it can also be seen that the model does not overfit to the training dataset as the training and validation errors closely follow each other.

The structure of the benchmark model is shown in Figure 5. The model consists of an input layer with three nodes, two

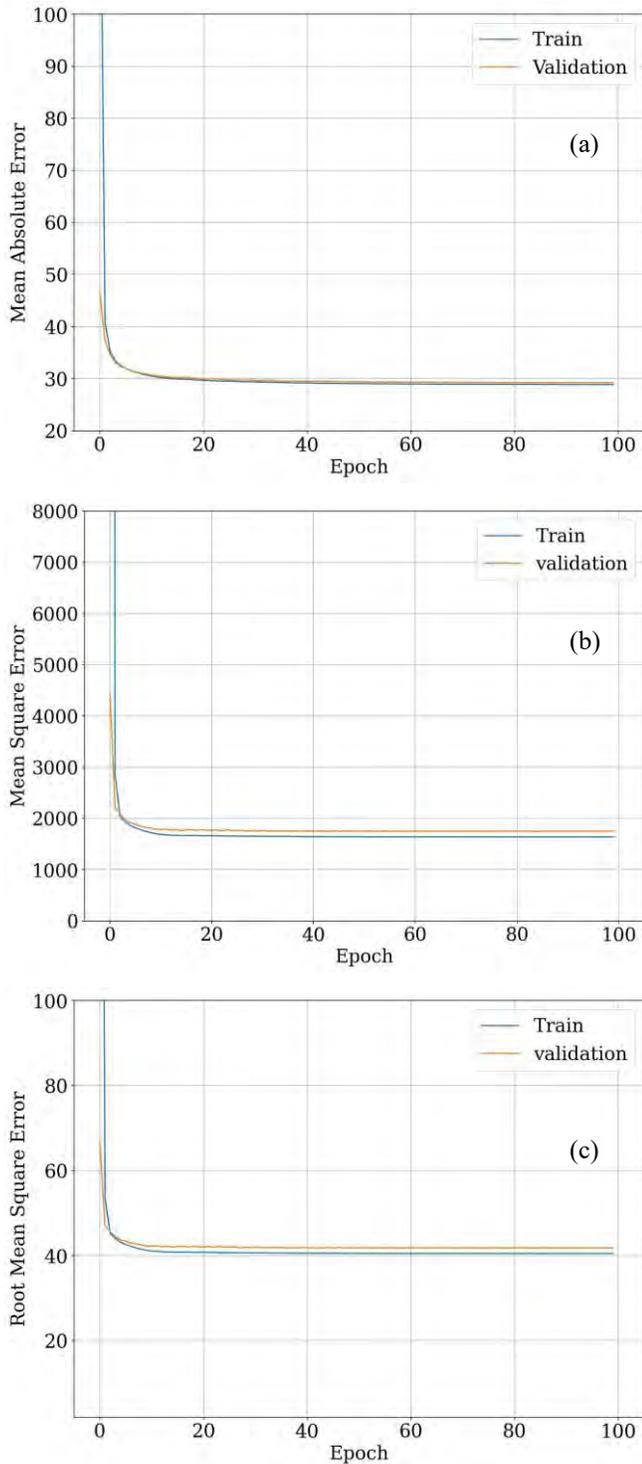


Figure 4. Variations in various error measures across epochs during the training process (a) MAE, (b) MSE, (c) RMSE

fully connected hidden layers with 16, and 32 neurons, respectively, and an output layer with a single neuron.

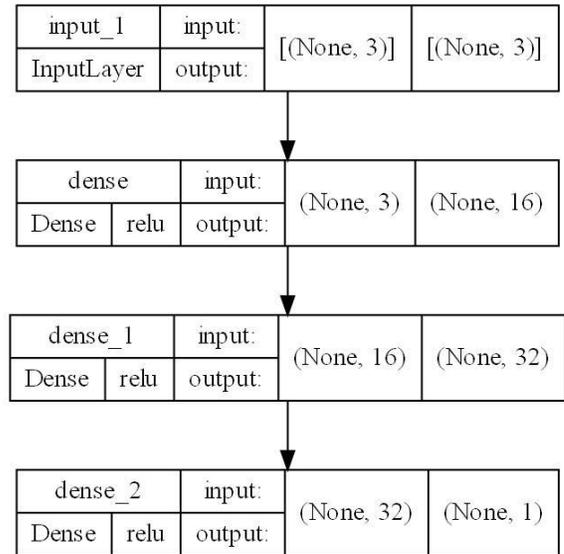


Figure 5. Architecture of the DNN benchmark model

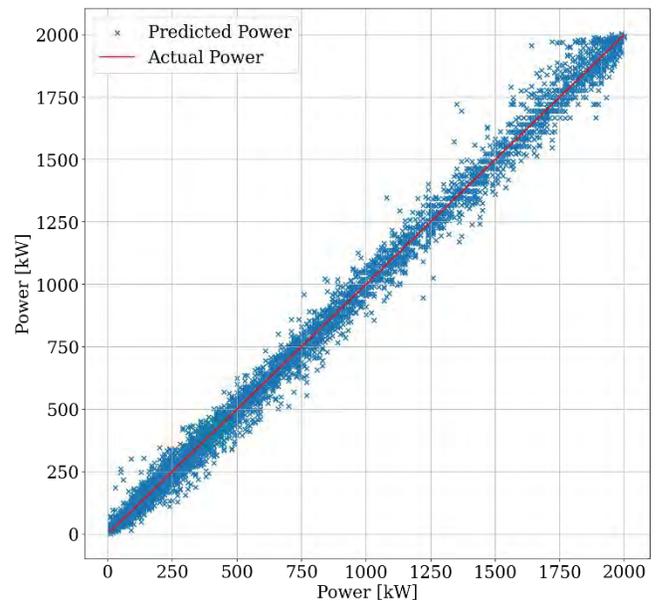


Figure 6. Comparison between the model predictions and the measured power

The model thus finalized was tested with the test dataset which was not seen by the model previously. The power predicted by the model (scattered points) is compared with the corresponding measured power (represented by the red line) from the turbine as represented in Figure 6. With an R squared value of 0.996, the proposed benchmark model could efficiently capture the performance variations of the turbine under various operating conditions. This is further evident in Figure 7, where the monthly averaged predictions and measurements over the study period are compared.

Performance of the benchmark model on the test dataset is further quantified with various error metrics like MAE,

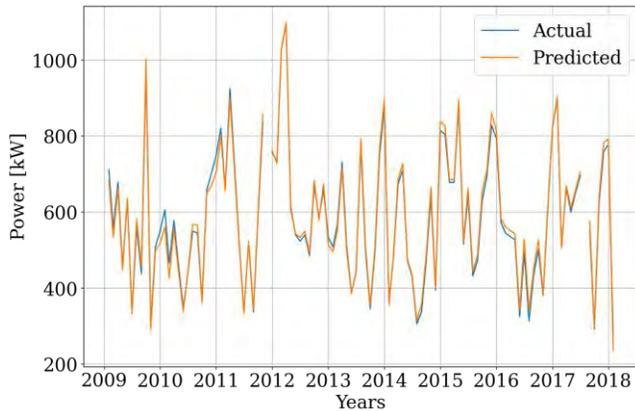


Figure 7. Comparison between the power predicted by the model and the actual power on monthly basis

RMSE, and MSE which were found to be 29.30, 41.37, and 1711.43 respectively.

In Veena et al. (2020), four different algorithms have been used to estimate the performance of a similar wind turbine of 2 MW size in a different location. Artificial neural network (ANN), support vector machine (SVM), k-nearest neighbors (k-NN), and multivariate adaptive regression splines (MARS) algorithms were used for this purpose. Even though the SVM performance is found to be better by the authors, the benchmark model developed using the DNN architecture outperforms all these algorithms. The MAE of the SVM model was found to be 91.10 while in our study, the MAE is 29.30. Thus, with a significant improvement in the monitored metrics, the use of DNN for model development is justified.

To explain the power prediction by the benchmark model, SHAP analysis has been conducted as discussed in the previous section. The results are shown through Figure 8 and Figure 9. As in the preliminary model, the output power prediction for the turbine is significantly contributed by the wind speed followed by generator RPM and yaw position. The explainability of this model helps in identifying the feature contributing to each individual prediction. Such an explanatory analysis, in contrast with the black box approach of the traditional AI models, could help windfarm operators and system managers in understanding the model better and thereby adopting it for decision making during the day-to-day operations. Further, explainable AI could also be used to identify problems the model may run into and can help in debugging the issues. With these high accuracies and explainability, the benchmark model was used for estimating the age-related performance degradation of the turbine as discussed in the next section.

5. EFFICIENCY INDEX

In most of the previous studies, age related performance decline of windfarms was estimated based on the data from several farms, considering the aggregated changes in

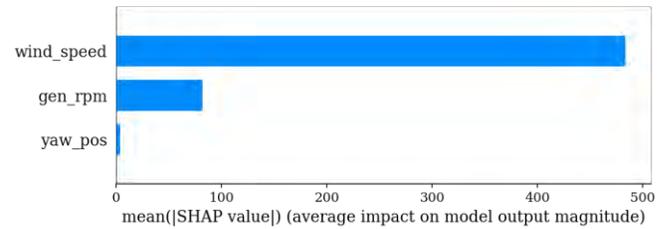


Figure 8. The average SHAP values for various input features of the benchmark model

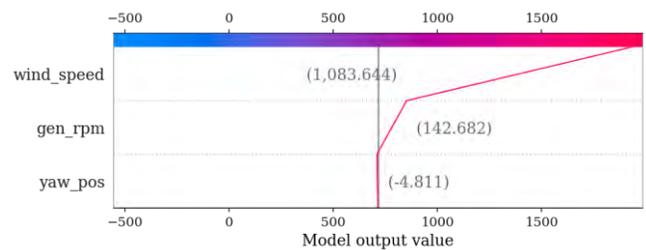


Figure 9. SHAP values for the preliminary model corresponding to a single sample

normalized capacity factor over the years. In this study on a specific turbine which is based on the high-resolution data collected from the SCADA system, the performance decline is measured through the time series drop in the turbine's overall efficiency. The power input from the wind to the turbine (P_{in}) is given by:

$$P_{in} = \frac{1}{2} \rho_a A_T V^3 \quad (1)$$

where ρ_a is the density of air, A_T is the area of the wind turbine's rotor and V is the incoming wind velocity. Whereas the power developed by the turbine at this wind velocity is given by:

$$P_T = C_P \eta_{tran} \eta_{gen} \frac{1}{2} \rho_a A_T V^3 \quad (2)$$

where C_P is the power coefficient of the turbine, η_{tran} is the combined efficiency of the drivetrain and η_{gen} is the generator efficiency.

Hence, the overall efficiency of the turbine η_T can be expressed as

$$\eta_T = \frac{P_T}{P_{in}} = \frac{C_P \eta_{tran} \eta_{gen} \frac{1}{2} \rho_a A_T V^3}{\frac{1}{2} \rho_a A_T V^3} = C_P \eta_{tran} \eta_{gen} \quad (3)$$

In this study, the age-related performance degradation of the turbine is proposed to be gauged through the time series decline in the efficiency index (η_I) of the turbine which is defined as

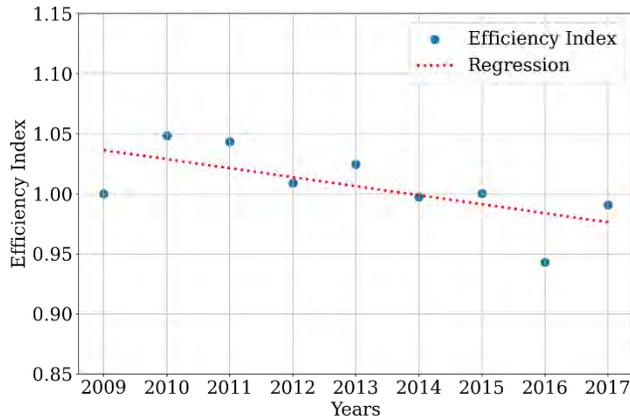


Figure 10. Decline in the efficiency index of the turbine over the years

$$\eta_I = \frac{\eta_{T_{Measured}}}{\eta_{T_{Modelled}}} \quad (4)$$

where $\eta_{T_{Measured}}$ is the overall efficiency of the system at a given wind velocity, estimated based on the actual measured power, and $\eta_{T_{Modelled}}$ is the corresponding system efficiency based on the power predicted by the model at the same wind velocity.

6. PERFORMANCE DEGRADATION PATTERN

The Power developed by the turbine during the years from 2009 to 2017 were predicted by running the benchmark model with the 10 min resolution input data corresponding to these years (2008 was not included due to excessive missing data points during this year). The data points were further reduced after the data filtering and pre-processing as discussed in section 2. The overall system efficiencies, corresponding to these model predictions ($\eta_{T_{Modelled}}$) were calculated using Eq. (3). Similarly, the efficiencies corresponding to the measured power ($\eta_{T_{Measured}}$) was also calculated. From these, the efficiency indexes (η_I) were computed using Eq. (4). These efficiency indexes were then averaged over different years and regressed as shown in Figure 10. It can be observed that the efficiency indices of the turbine decline over the years, indicating the age-related performance degradation of the system. One of the major causes of this performance degradation could be the gradual wear and tear of the turbine’s power transmission components and generator.

However, these issues are usually resolved during the maintenance of the system. Another significant reason for this could be erosion of the rotor blade tips, which will

adversely affect the aerodynamic performance of the rotor and thereby the power coefficient, C_p . In this context, it is worth mentioning that the site at which the turbine is installed is exposed to extreme weather events like heavy rain and snow, which will aggravate the tip erosion of the blades.

The annual average decline in the efficiency index of the turbine is 0.64 %. This value is between the reported average degradation of 0.53% per year for the US systems (Hamilton et al., 2020) and 0.87 % for turbines in the UK (Hughes, 2012). However, the degradation rate of this Norwegian turbine is less than the reported rate of 1.6% per year under another UK based study by Staffell and Green (2014). These differences in the reported degradation rates could be due to the variations in the environments under which the systems are exposed while in operation.

7. CONCLUSIONS

In this paper, we analyze the age-related performance degradation of a 2 MW wind turbine, working under the Norwegian environment. Ten years of high-resolution operational data from the turbine were used for this analysis in which the actual performance of the turbine over the years were benchmarked with the performance of the system modelled based on its initial year’s performance data. For this, a DNN based preliminary performance model was developed which was interpreted through SHAP analysis. Based on this, an efficient benchmark model for the turbine’s performance was developed with an optimized DNN architecture. An efficiency index has been proposed to estimate the age-related performance decline of the turbine from its expected benchmarks. The age-related performance degradation of the turbine is evident from the declining trend of the efficiency index over the years of operation. On an average, the efficiency index of the turbine was found to decline by 0.64 percent every year of its operation. In spite of the slight differences in the degradation rates, the current estimates on the performance decline of the Norwegian turbine are comparable with the results from similar studies on the US and the UK based turbines. The results of the study can give some useful indications for the timely interventions for performance enhancement of the turbine through appropriate overhauling and refurbishing. Further, the analysis can be extended for the estimation of the Remaining Useful Life (RUL) of wind turbines using efficient Recurrent Neural Network architectures like Long Short-Term Memory (LSTM).

ACKNOWLEDGEMENTS

This research work has been funded by Analytics for asset Integrity Management of Windfarms (AIMWind), under grant no. 312486, from Research Council of Norway (RCN). AIMWind is collaborative research from University of Agder, Norwegian Research Center (NORCE), and TU Delft, with DNV and Origo Solutions as advisory partners. Support

from Jørgen Olsen, Statkraft is also thankfully acknowledged.

REFERENCES

- Adedipe, T., & Shafiee, M. (2021). An economic assessment framework for decommissioning of offshore wind farms using a cost breakdown structure. *The international journal of life cycle assessment*, 26(2), 344-370. <https://doi.org/10.1007/s11367-020-01793-x>
- Astolfi, D., Byrne, R., & Castellani, F. (2021). Estimation of the performance aging of the vestas v52 wind turbine through comparative test case analysis. *Energies (Basel)*, 14(4), 915. <https://doi.org/10.3390/en14040915>
- Astolfi, D., Castellani, F., Lombardi, A., & Terzi, L. (2021). Data-driven wind turbine aging models. *Electric Power Systems Research*, 201, 107495. <https://doi.org/10.1016/j.epsr.2021.107495>
- Byrne, R., Astolfi, D., Castellani, F., & Hewitt, N. J. (2020). A study of wind turbine performance decline with age through operation data analysis. *Energies (Basel)*, 13(8), 2086. <https://doi.org/10.3390/en13082086>
- Dai, J., Yang, W., Cao, J., Liu, D., & Long, X. (2018). Ageing assessment of a wind turbine over time by interpreting wind farm scada data. *Renewable Energy*, 116, 199-208. <https://doi.org/10.1016/j.renene.2017.03.097>
- Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *kdd*,
- Germer, S., & Kleidon, A. (2019). Have wind turbines in germany generated electricity as would be expected from the prevailing wind conditions in 2000-2014? *PLoS One*, 14(2), e0211028-e0211028. <https://doi.org/10.1371/journal.pone.0211028>
- Global Wind Energy Council. (2021). *Global wind report*. Council, G. W. E.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.
- Hamilton, S. D., Millstein, D., Bolinger, M., Wiser, R., & Jeong, S. (2020). How does wind project performance change with age in the united states? *Joule*, 4(5), 1004-1020. <https://doi.org/https://doi.org/10.1016/j.joule.2020.04.005>
- He, K., Zhang, X., Ren, S., & Sun, J. (2015, 2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.
- Hughes, G. (2012). The performance of wind farms in the united kingdom and denmark. *Renewable Energy Foundation*, 48.
- International Renewable Energy Agency. (2019). Future of wind: Deployment, investment, technology, grid integration and socio-economic aspects. *Abu Dhabi*.
- Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 31(3), 685-695.
- Kim, H.-G., & Kim, J.-Y. (2021). Analysis of wind turbine aging through operation data calibrated by lidar measurement. *Energies (Basel)*, 14(8), 2319. <https://doi.org/10.3390/en14082319>
- Lundberg, S., & Lee, S.-I. (2017). A unified approach to interpreting model predictions.
- Olauson, J., Edström, P., & Rydén, J. (2017). Wind turbine performance decline in sweden. *Wind energy (Chichester, England)*, 20(12), 2049-2053. <https://doi.org/10.1002/we.2132>
- Pan, Y., Hong, R., Chen, J., Feng, J., & Wu, W. (2021). Performance degradation assessment of wind turbine gearbox based on maximum mean discrepancy and multi-sensor transfer learning. *Structural Health Monitoring*, 20(1), 118-138. <https://doi.org/10.1177/1475921720919073>
- Sareen, A., Sapre, C. A., & Selig, M. S. (2014). Effects of leading edge erosion on wind turbine blade performance: Effects of leading edge erosion. *Wind energy (Chichester, England)*, 17(10), 1531-1542. <https://doi.org/10.1002/we.1649>

- Somers, M. J., & Casal, J. C. (2009). Using artificial neural networks to model nonlinearity: The case of the job satisfaction—job performance relationship. *Organizational Research Methods*, 12(3), 403-417.
- Staffell, I., & Green, R. (2014). How does wind farm performance decline with age? *Renewable Energy*, 66, 775-786. <https://doi.org/10.1016/j.renene.2013.10.041>
- Sulaiman, M. S., Abood, M. M., Sinnakaudan, S. K., Shukor, M. R., You, G. Q., & Chung, X. Z. (2021). Assessing and solving multicollinearity in sediment transport prediction models using principal component analysis. *ISH Journal of Hydraulic Engineering*, 27(S1), 343-353. <https://doi.org/10.1080/09715010.2019.1653799>
- Veena, R., Mathew, S., & Petra, M. I. (2020). Artificially intelligent models for the site-specific performance of wind turbines. *International Journal of Energy and Environmental Engineering*, 11(3), 289-297. <https://doi.org/10.1007/s40095-020-00352-2>
- Wiser, R. H., & Bolinger, M. (2019). Benchmarking anticipated wind project lifetimes: Results from a survey of u.S. Wind industry professionals.
- Ziegler, L., Gonzalez, E., Rubert, T., Smolka, U., & Melero, J. J. (2018). Lifetime extension of onshore wind turbines: A review covering germany, spain, denmark, and the uk. *Renewable & sustainable energy reviews*, 82, 1261-1271. <https://doi.org/10.1016/j.rser.2017.09.100>
- Zweiri, F., Vanna, F. D., Heidari, A., Benini, E., Williams, N., & Hadavinia, H. (2022, 03/02/2022). *The effect of leading edge erosion on wind turbine blade aerodynamic performance* 3rd International Symposium on Leading Edge Erosion of Wind Turbine Blades, Denmark.

BIOGRAPHIES

Manuel S. Mathew is a PhD Research Fellow at the Information and Communication Technology department at the University of Agder, Norway. His interest is in application of artificial intelligence in renewable energy systems particularly focusing on prognostics for wind farms. He completed his master's degree in Renewable Energy in 2021 from the University of Agder. In addition, he also holds a master's degree in Systems Engineering by research from the University of Brunei Darussalam. He did his bachelor's degree in Electrical and Electronics Engineering from the Mahatma Gandhi University, India.

Surya Teja Kandukuri is a Senior Data Scientist at Cognite AS. He holds a part-time position as post-doctoral research fellow at University of Agder, Grimstad, Norway. He obtained PhD in condition monitoring from University of Agder in 2018. He has over 12 years of experience in industrial research within aerospace, energy, marine and oil & gas sectors, developing condition monitoring solutions for high-value assets. He received his master's degree in systems and control engineering from TU Delft, The Netherlands, in 2006 and bachelor's in electrical engineering from Nagarjuna University in India in 2003.

Christian W. Omlin has been a professor of Artificial Intelligence at the University of Agder since 2018. He has previously taught at the University of South Africa, University of the Witwatersrand, Middle East Technical University, University of the South Pacific, University of the Western Cape, and Stellenbosch University. His expertise is in deep learning with a focus on applications ranging from safety to security, industrial monitoring, renewable energy, banking, sign language translation, healthcare, bio conservation, and astronomy. He is particularly interested in the balance between the desire for autonomy using AI technologies and the necessity for accountability through AI imperatives such as explainability, privacy, security, ethics, and artificial morality for society's ultimate trust in and acceptance of AI. He received his Ph.D. from Rensselaer Polytechnic Institute and his MEng from the Swiss Federal Institute of Technology, Zurich, in 1995 and 1987, respectively.

Weighted-QMIX-based optimization for maintenance decision-making of multi-component systems

Van-Thai Nguyen, Phuc Do, Alexandre Voisin, and Benoit Iung

Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France

van-thai.nguyen@univ-lorraine.fr

phuc.do@univ-lorraine.fr

alexandre.voisin@univ-lorraine.fr

benoit.iung@univ-lorraine.fr

ABSTRACT

It is well-known that maintenance decision optimization for multi-component systems faces the curse of dimensionality. Specifically, the number of decision variables needed to be optimized grows exponentially in the number of components causing computational expensive for optimization algorithms. To address this issue, we customize a multi-agent deep reinforcement learning algorithm, namely Weighted QMIX, in the case where system states can be fully observed to obtain cost-effective policies. A case study is conducted on a 13-component system to examine the effectiveness of the customized algorithm. The obtained results confirmed its performance.

1. INTRODUCTION

Maintenance policy can be classified into two main categories, namely, corrective and preventive maintenance (CM and PM) (H. Wang, 2002). CM carries out maintenance actions on failed machines, which is often associated with high related costs due to unexpected production losses as well as unscheduled maintenance costs (Ahmad & Kamaruddin, 2012). On the contrary, PM implements maintenance on functioning machines to prevent their sudden failures in order to reduce downtime costs (Huang, Chang, & Arinez, 2020).

PM interventions can be planned in either time-oriented or condition-based manner (CBM), however, the later appears to be more advantageous. Particularly, it allows flexibly selecting maintenance decisions based on current states of maintained machines instead of on a fixed scheduled calendar. Moreover, recent advances in sensing and information technology allows rich degradation data to be collected enabling CBM to become a popular approach for maintenance decision-

making and optimization.

CBM policies can be divided into two main groups: direct mapping and threshold-based policy. While the former maps directly from component degradation measurements to maintenance actions, the later first compares component states to predefined thresholds, and then choose maintenance actions accordingly. Whereas CBM optimization processes used for single-unit systems can be effectively achieved due to the small number of decision variables needed to be optimized (Quatrini, Costantino, Di Gravio, & Patriarca, 2020), the ones of multi-component systems suffer from the curse of dimensionality. Particularly, the number of decision variables grows rapidly as the number of components increases, causing computational expensive for optimization algorithms (Zhang & Si, 2020).

Recent advancements in the field of reinforcement learning (RL) give rise to direct mapping approaches by providing new tools for single-agent deep RL algorithms (DRL) to deal with maintenance decision optimization of systems with large state spaces. Specifically, (Zhang & Si, 2020) used double deep Q-network (DDQN) algorithm to minimize cost for a 12-component system which suffers from stochastic, economic dependence and competing failure risks. DDQN is also employed to optimize maintenance cost for systems with extremely large state spaces showing better performance in comparison to threshold-based policies (Huang et al., 2020). Despite the success of single-agent DRL algorithms for maintenance applications with state space complexities, they are shown in the literature to suffer from the problem of large action spaces (Andriotis & Papakonstantinou, 2019). In particular, the output layer of a conventional deep Q-network is composed of Q-value for each available action, and is then equal to the action space's size. Similarly, actor networks of policy-gradient-based DRL algorithms output a probabilistic distribution over all possible actions. As a result, these network structures of single-agent DRL algorithms are not

Van-Thai Nguyen et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

suitable for applications with sizable action spaces.

Fortunately, the framework of multi-agent DRL (MADRL) appears as a promising solution to this challenge. Particularly, in a multi-agent maintenance planning task, each agent observes the state of a subsystem or the system and determines maintenance actions of one or more components. As a result, the action space of an agent is much smaller than the one of the entire system, which helps alleviate the issue of large action spaces at system level (Zhou, Li, & Lin, 2021). MADRL has been received recently increasing attention from maintenance researchers and several related articles are reviewed in the following. Wolpertinger deep deterministic policy gradient algorithm is employed in (Liu, Chen, & Jiang, 2019) to optimize selective maintenance policies for a coal transport system consisting of 14 components. However, this algorithm requires a nearest neighbor layer used for action reduction that interrupts the differentiability of the network, potentially leading to training instabilities due to improper backpropagation of gradients (Andriotis & Papakonstantinou, 2019). Deep centralized multi-agent actor-critic (DCMAC) algorithm is developed by (Andriotis & Papakonstantinou, 2019) to optimize maintenance actions for large structures. The truncated importance sampling mechanism is employed in DCMAC to cope with high variance in gradient estimators of learning policies, however, bias still exists which may cause unstable training (Z. Wang et al., 2016). More recently, hierarchical coordinated DRL algorithm is proposed in (Zhou et al., 2021) to optimize maintenance decisions of a specific natural gas plant consisting of 14 components which may be difficult to be applied for other kinds of systems.

Besides, recent advances in monotonically decomposing joint action-value functions allow to improve MADRL algorithms' scalability as well as training stability. Among the papers employing this technique, Weighted QMIX (WQMIX) (Rashid, Farquhar, Peng, & Whiteson, 2020) is one of the state-of-the-art algorithms, however, its performance for maintenance decision-making has not been investigated yet. Furthermore, WQMIX adopts the centralized training and decentralized execution paradigm which may cause slow learning in applications where agents can fully observe system states.

To address these issues, in this paper, we customize WQMIX to effectively optimize maintenance decisions of large-scale multi-component systems for the fully observable setting. In particular, separate agent networks are replaced by a single branching dueling network (branching network) (Tavakoli, Pardo, & Kormushev, 2018) to take advantage of the fully observable setting. The branching structure allows achieving a linear increase in the size of deep Q-networks's output layer to avoid the curse of dimensionality. Moreover, it also allows to create virtual communication channels between learning agents to facilitate decision-making processes as well as to avoid the use of recurrent neural networks in agent networks

in the original paper that may slow down learning processes.

Our main contributions in this study are two folds. Firstly, we customize WQMIX algorithm specifically for the fully observable setting. Secondly, we conduct a comparison study to benchmark the performance of the customized algorithm, the branching dueling deep Q-learning (Tavakoli et al., 2018) and a threshold-based policy when they are used to optimize maintenance actions of large-scale systems.

The rest of the paper is organized as following. Section 2 is devoted to the general description of the maintained system. Maintenance operations and optimization problem statement formulation are described in section 3. The fully cooperative multi-agent setting for maintenance decision-making is depicted in section 4 and the detail of maintenance optimization process is presented in section 5. The numerical results are depicted and analyzed in section 6. The conclusions drawn from this work and some perspectives are presented in the last section.

2. SYSTEM DESCRIPTION

We consider a series-parallel system being composed of N components which can be grouped into M subsystems. It is assumed that subsystem i contains H^i components of the same type i . As a result, $N = \sum_{i=1}^M H^i$.

A component of type i at a periodical inspection time t_k can be observed in any discrete health state, $s_k^i \in \{0, \dots, m^i\}$, ranging from new to complete failure. Furthermore, it is also assumed that without any maintenance intervention, the state transition of a component of type i between two successive inspections obeys its inherent Markov probability transition matrix that has the following form:

$$P^i = \begin{bmatrix} p_{00}^i & p_{01}^i & p_{02}^i & \cdots & p_{0m_i}^i \\ 0 & p_{11}^i & p_{12}^i & \cdots & p_{1m_i}^i \\ 0 & 0 & p_{22}^i & \cdots & p_{2m_i}^i \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (1)$$

in which p_{uv}^i is a non-negative real number representing the degradation transition probability from state u to state v of a component of type i that satisfies: $\sum_{v=u}^{m^i} p_{uv}^i = 1, \forall u \in \{0, \dots, m^i\}$.

As an example, figure 1 illustrates a 13-component series system with four parallel subsystems. Specifically, component 1 is considered as the first subsystem. Component 2, 3 and 4 together form the next subsystem. The third one is composed of component 5, 6, 7 and 8. The last subsystem consists of the remaining components.

The degradation transition matrices of the four component

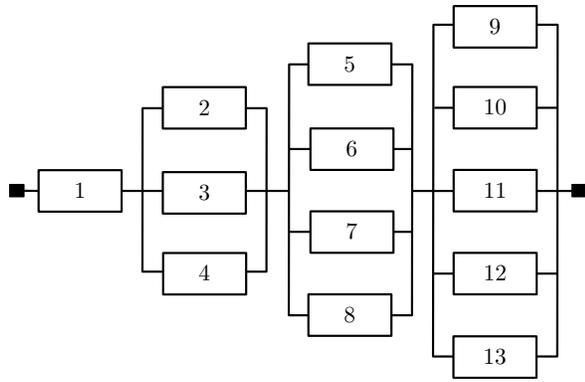


Figure 1. Reliability block diagram of the studied system

types are given as follows:

$$P^1 = \begin{bmatrix} 0.60 & 0.30 & 0.05 & 0.05 \\ 0.00 & 0.60 & 0.30 & 0.10 \\ 0.00 & 0.00 & 0.60 & 0.40 \\ 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix} \quad (2)$$

$$P^2 = \begin{bmatrix} 0.50 & 0.30 & 0.10 & 0.10 \\ 0.00 & 0.50 & 0.30 & 0.20 \\ 0.00 & 0.00 & 0.50 & 0.50 \\ 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix} \quad (3)$$

$$P^3 = \begin{bmatrix} 0.65 & 0.25 & 0.05 & 0.05 \\ 0.00 & 0.65 & 0.25 & 0.10 \\ 0.00 & 0.00 & 0.65 & 0.35 \\ 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix} \quad (4)$$

$$P^4 = \begin{bmatrix} 0.60 & 0.30 & 0.05 & 0.05 \\ 0.00 & 0.60 & 0.30 & 0.10 \\ 0.00 & 0.00 & 0.60 & 0.40 \\ 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix} \quad (5)$$

3. MAINTENANCE OPERATIONS

Maintenance operations of the studied system are planned following both CM and PM strategy. A CM intervention replaces a failed component by a new one of the same type. PM actions could be either perfect or imperfect. While a perfect PM action completely restores a survival component to be as new, the imperfect one implies that state after maintenance of a component is somewhere between its state before maintenance and “as good as new” state. Moreover, it should be noted that maintenance actions can only be carried out after component states are revealed by inspections. As a result, when the failure of a component or a group of components occurs between two consecutive inspections, maintenance actions must wait until next scheduled inspection to be implemented.

The cost of maintaining a component individually consists of an inspection cost, a setup cost and a component-specific maintenance cost. The inspection cost is denoted as c^{ins} , which is necessarily paid even for survival components in order to reveal their current degrading states. We consider in this paper two levels of setup cost following (Wijnmalen & Hontelez, 1997), which are system setup cost, c^0 , caused by, for example, transportation of spare parts or administrative handling, and component-type setup cost, $c^{t,i}$, originated from the requirement of specific tools or repairman skills. The component-specific maintenance cost is denoted as $c^{m,i}$ which depends on maintenance quality. Based on the above descriptions, the cost of separately maintaining a component i is computed as follows:

$$c^i = c^0 + c^{t,i} + c^{ins} + c^{m,i} \quad (6)$$

In practice, maintenance operations of multi-component systems usually benefits from shared setup costs when several components are grouped to maintain thanks to the positive economic dependency between them. For the studied system, the system setup cost can only be charged once if several components are maintained together. In the same manner, the component-type setup costs are charged once if a group of components of the same type are maintained simultaneously. As a result, the total maintenance cost at system level denoted as c can be calculated as follows:

$$c = \sum_{i=1}^N c^i - \mathbb{I}^0(N^0 - 1)c^0 - \sum_{i=1}^M \mathbb{I}^{m,i}(H^{m,i} - 1)c^{t,i} \quad (7)$$

in which N^0 is the number of maintained components; $H^{m,i}$ is the number of maintained components of subsystem i ; \mathbb{I}^0 is the system maintenance indicator whose value is equal to one if there is at least one component being maintained or equal to zero otherwise; $\mathbb{I}^{m,i}$ is the maintenance indicator of type i whose value is equal to one if there is at least one component of type i being maintained or equal to zero otherwise. In addition to the maintenance cost, the downtime cost denoted as c^{dt} that is caused by the failure of a component or a group of components leading to the shutdown of the system should be considered. Our objective of maintenance decision-making optimization is to minimize to the long-run average cost rate.

4. FULLY COOPERATIVE MULTI-AGENT SETTING FOR MAINTENANCE DECISION-MAKING

4.1. Agent-environment interaction

The maintenance optimization problem of the studied system is modeled as a fully cooperative multi-agent decision-making task with a group of N agents, $\mathcal{AG} = \{AG^i\}_{i=1}^N$, in which one agent controls maintenance decisions of one component and can fully observe system states. A component i has its own state space, \mathcal{S}^i , that help form the state space at

system level, $\mathcal{S}^{joint} \equiv \mathcal{S}^1 \times \mathcal{S}^2 \times \dots \times \mathcal{S}^N$. At any inspection time t_k , each agent $AG^i \in \mathcal{AG}$ observes the system's current state $\mathbf{s}_k \in \mathcal{S}^{joint}$ and then choose an action a_k^i from its own action space, \mathcal{A}^i , based on the observation and its own policy denoted as π^i . The actions chosen by individual agents form a joint action $\mathbf{a}_k \in \mathcal{A}^{joint} \equiv \mathcal{A}^1 \times \mathcal{A}^2 \times \dots \times \mathcal{A}^N$. Once the chosen joint action is implemented, the system transitions to a state after maintenance $\bar{\mathbf{s}}_k$ and releases a numerical reward r_k shared by all agents. After that, it degrades naturally to a next state before maintenance \mathbf{s}_{k+1} at inspection time t_{k+1} according to the transition matrices \mathbf{P}^i ($i = \{1, \dots, M\}$).

4.1.1. Environment element definition

The definition of system state, action and reward distribution mechanism within the context of this study is presented in the following paragraphs.

System state The state of a component i at inspection time t_k is its degradation level s_k^i . Hence, the system state at that time is a vector consisting all component states defined as $\mathbf{s}_k = [s_k^1, s_k^2, \dots, s_k^N]^T$.

System action Similarly, the action at system level at inspection time t_k is a vector being composed of all component maintenance actions which is mathematically defined as: $\mathbf{a}_k = [a_k^1, a_k^2, \dots, a_k^N]^T$ where a_k^i is the maintenance action of component i at that time. More specifically, there are three possible maintenance actions for each component which are encoded as follows:

$$a_k^i = \begin{cases} 0 & \text{leave component } i \text{ as it is} \\ 1 & \text{perform imperfect maintenance on component } i \\ 2 & \text{replace component } i \text{ by the one of the same type} \end{cases} \quad (8)$$

Imperfect maintenance implies that a component can be maintained to be in a better state which is some where between its current state and “as good as new”. We employ the imperfect maintenance model in (Do & Bérenguer, 2012) where state after maintenance of a component can be obtained by sampling uniformly discretely from the interval from new state to its state before maintenance. The cost of maintaining a component i is computed as follows:

$$c_k^{m,i} = c_k^{r,i} \cdot \left(\frac{s_k^i - \bar{s}_k^i}{s_k^i} \right)^\beta \quad (9)$$

in which $c_k^{r,i}$ is a constant representing the replacement cost of component i ; s_k^i and \bar{s}_k^i are respectively the state before and after maintenance of component i ; β is a real positive number representing the components' imperfect maintenance characteristics.

Reward Maintenance optimization involves balancing the trade-off between maintenance frequency and downtime cost, which means that if maintenance is conducted too often, maintenance cost can be high or if maintenance operations are conducted less frequently, downtime cost is more prone to occurrence. To deal with this issue, the reward function used in this work is defined as the opposite of the total cost which is the sum of the maintenance cost at system level and the downtime cost as follows:

$$r_k = -c_k - \mathbb{I}_k c^{dt} \quad (10)$$

in which \mathbb{I}_k is the system failure status indicator at time t_k whose value is equal to one if the system is in failed state at that time or is equal to zero otherwise; c^{dt} is a real constant representing downtime cost.

5. MADRL-BASED MAINTENANCE OPTIMIZATION

We first present BDQ algorithm to introduce the branching network in subsection 5.1 and then the customized WQMIX algorithm in subsection 5.2.

5.1. Branching dueling Q-learning (BDQ)

DRL has emerged recently as an effective framework for solving decision-making tasks with large state spaces by using deep neural networks to approximate action-value functions. This parameterized functional form allows to reduce the problem of determining values at each point in a Q-table to determining the number of weights of the corresponding network which is much less than the number of state-action pairs (Andriotis & Papakonstantinou, 2019). Moreover, the weight sharing in neural networks enables the generalization in the sense that updating weights for a single state-action pair affect the estimation of action values of other state-action pairs. Despite the success of DRL algorithms for applications with state space complexities, they may not be efficient for the ones with large action spaces due to the fact that the output layer of a deep Q-network or a dueling deep-Q network consist of Q-values for each available actions. As a result, its size is equal to the size of action space (Andriotis & Papakonstantinou, 2019). For the system studied in this paper, the size of action space is equal to $\prod_{i=1}^N |\mathcal{A}^i|$ which grows exponentially in the number of components.

To tackle this problem, the BDQ algorithm in (Tavakoli et al., 2018) provides a special network structure, namely, branching dueling deep Q-network (or branching network for short), that allows the number of outputs of deep Q-networks to linearly increases with the number of components as illustrated in figure 2.

Specifically, at component level, each agent AG^i chooses a maintenance decision $a_k^i = \operatorname{argmax}_{a_k^i \in \mathcal{A}^i} Q^i(\mathbf{s}_k, a_k^i)$ based on its own action-value function which is computed based on its own advantage function, $\Omega^i(\mathbf{s}_k, a_k^i)$ and the state-value

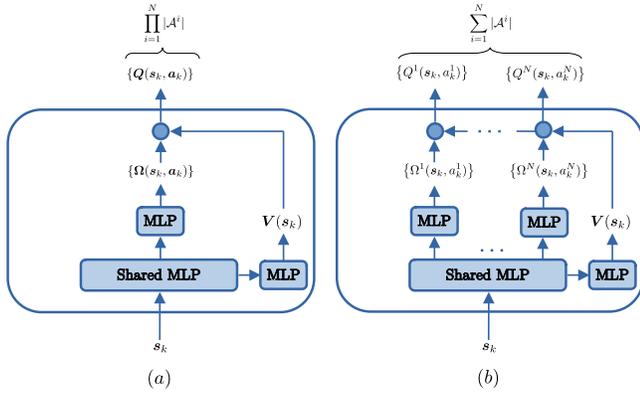


Figure 2. (a) Dueling Q-network structure. (b) Branching dueling Q-network structure.

function that is shared by all agents, $V(s_k)$, as belows:

$$Q^i(s_k, a_k^i) = V(s_k) + \Omega^i(s_k, a_k^i) - \frac{1}{|\mathcal{A}^i|} \sum_{a_k \in \mathcal{A}^i} \Omega^i(s_k, a_k^i) \quad (11)$$

At system level, the agents cooperatively resolve to select a joint maintenance action a_k which is defined in previous section as a vector of all component maintenance actions according to the following equation:

$$a_k = \begin{bmatrix} \operatorname{argmax}_{a_k^1 \in \mathcal{A}^1} Q^1(s_k, a_k^1) \\ \operatorname{argmax}_{a_k^2 \in \mathcal{A}^2} Q^2(s_k, a_k^2) \\ \dots \\ \operatorname{argmax}_{a_k^N \in \mathcal{A}^N} Q^N(s_k, a_k^N) \end{bmatrix} \quad (12)$$

The loss of one transition sample, (s_k, a_k, r_k, s_{k+1}) , used to train the branching network is aggregated between all branches as belows:

$$L = \frac{1}{N} \sum_{i=1}^N (Q^i(s_k, a_k^i) - y)^2 \quad (13)$$

in which $y = r + \gamma \frac{1}{N} \sum_{i=1}^N Q_{target}^i(s_{k+1}, a_{k+1}^{i,*})$ is considered as the target which is shared between all branches where $a_{k+1}^{i,*} = \operatorname{argmax}_{a_{k+1}^i \in \mathcal{A}^i} Q^i(s_{k+1}, a_{k+1}^i)$. It should be noted that Q_{target}^i is computed from a separate branching network called “target network” whose weights are periodically copied from the one used to calculate Q^i after every fixed number of training steps.

5.2. The customized WQMIX

Despite the advantage of the BDQ’s branching network that allows to deal with the exponential increase in the number of outputs of deep Q-networks for high-dimensional systems, its training scheme is based on the idea of distributing temporal-

difference errors across all branches, which is a heuristic approach and lacks of theoretical methodology. Indeed, BDQ does not guarantee one of the most important concepts of multi-agent systems which is the decision selection consistency between component and system level in the sense that learning agents cooperate with each other to choose a joint maintenance action a_k according to the system action-value function, $Q(s_k, a_k)$, that should be consistent with actions chosen by local agents. The action selection consistency is mathematically expressed as:

$$\operatorname{argmax}_{a_k \in \mathcal{A}} Q(s_k, a_k) = \begin{bmatrix} \operatorname{argmax}_{a_k^1 \in \mathcal{A}^1} Q^1(s_k, a_k^1) \\ \operatorname{argmax}_{a_k^2 \in \mathcal{A}^2} Q^2(s_k, a_k^2) \\ \dots \\ \operatorname{argmax}_{a_k^N \in \mathcal{A}^N} Q^N(s_k, a_k^N) \end{bmatrix} \quad (14)$$

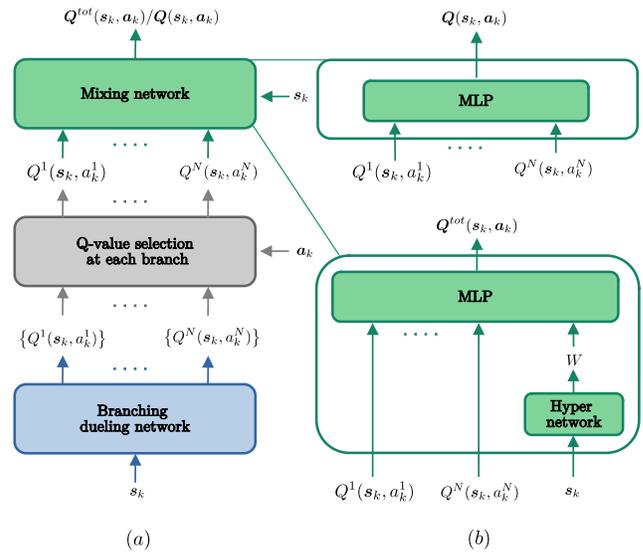


Figure 3. (a) The customized WQMIX architecture. (b) Mixing network structures.

Fortunately, the action selection consistency between component and system level can be achieved through the factorization method of VDN (Sunehag et al., 2017) which supposes that the system Q-function can be approximated by the sum of the per-component ones:

$$Q(s_k, a_k) \approx Q^{tot}(s_k, a_k) = \sum_{i=1}^N Q^i(s_k, a_k^i) \quad (15)$$

QMIX (Rashid et al., 2018) generalizes the VDN’s linear representation by assuming that $Q^{tot}(s_k, a_k)$ is a monotonic continuous function of the per-agent Q-functions, in other words, $\partial Q^{tot}(s_k, a_k) / \partial Q^i(s_k, a_k^i) \geq 0, \forall i \in \{1, \dots, N\}$. This assumption can be realized by using a multi-layer per-

ception (MLP) called mixing network that takes agents' own Q-values as inputs and outputting values of $Q^{tot}(s_k, a_k)$, whose weights are generated by a hyper network (Ha, Dai, & Le, 2016) to assure their values greater than or equal to zero.

The monotonic factorization scheme restricts an agent choosing its own actions independent of the actions chosen by other agents which may lead to the finding of suboptimal policies for applications required strong cooperation efforts between learning agents as in the case of maintenance decision-making for multi-component systems. (Rashid et al., 2020) showed that this factorization limit originates from the equal weighting placed on each joint action in the loss function used to update the joint Q-function, and proposed a weighting scheme to cope with this issue. Specifically, a MLP without any restriction to its weights is used to estimate system action values which is considered as baselines to put more attention on potential optimal joint actions in the loss function.

The network architectures used for computing $Q(s_k, a_k)$ and $Q^{tot}(s_k, a_k)$ is illustrate in figure 3. The losses of one transition sample, (s_k, a_k, r_k, s_{k+1}) , used to update the weights of these networks are computed as follows:

$$\begin{aligned} L^{Q^{total}} &= w(s_k, a_k)(Q^{total}(s_k, a_k) - y)^2 \\ L^Q &= (Q(s_k, a_k) - y)^2 \end{aligned} \quad (16)$$

where:

- $y = r + \gamma Q_{target}(s_{k+1}, \text{argmax}_{a_{k+1}} Q^{tot}(s_{k+1}, a_{k+1}))$ is the fixed target with Q_{target} is computed from the target networks of Q .
- $w(s_k, a_k)$ is the weight of joint action a_k whose value is equal to 1 if $Q^{tot}(s_k, a_k) < y$ or equal to $\alpha \in (0, 1]$ otherwise.

6. NUMERICAL STUDIES

This section compares the performance of the customized WQMIX with BDQ and a threshold-based policy for maintenance optimization of the 13-component system depicted in figure 1.

6.1. System parameters

All cost parameters are given in arbitrary units (acu) which are presented in the following. The inspection cost c^{ins} and system setup cost c^0 are 5 and 30 (acu) respectively. The component setup cost of type i is $c^{t,i} \in \{25, 20, 15, 10\}$ for $i = 1, 2, 3, 4$. The replacement costs $c^{r,i}$ of four component types are 65, 60, 55, 50 (acu) respectively. The downtime cost constant is $c^{dt} = 1000$ (acu). Finally, the imperfect maintenance parameter β is set to 3.

6.2. Training descriptions

The branching network of WQMIX and BDQ takes component states as input, hence, its input layer's size is 13. The shared MLP consists of two layers of 128 hidden units and the advantage MLP for each branch and the MLP used for computing system value function are composed of a single layer of 64 and 128 hidden units respectively. The number of outputs in each branch is equal to the number of maintenance actions at component level which is 3. The mixing network of Q^{tot} consists of two hidden layers of 64 units, whose weights are generated by a two separate hyper-networks of 64 units. The mixing network of Q is a MLP of two hidden layers of 64 units.

It should be noted that due to the maintenance constrain described in section 3 that if a component is failed, it can only be replaced by a new one of the same type or be left as it is, we classify a component action, a_k^i , chosen at a given system state s_k^i as a wrong action if $s_k^i = m^i$ then $a_k^i = 1$, or as a feasible action, otherwise. In order to realize this constrain, an action mask is applied to filter out invalid actions. In particular, Q-value corresponding to wrong actions at each branch are forced to be $-\infty$ to guarantee that invalid actions cannot be chosen by DRL agents.

The two MADRL algorithms are trained through 2×10^6 steps. Learning rates are scheduled to decline from 10^{-3} to 0.25×10^{-3} during the first 300×10^3 training steps. Through training, exploration constant is annealed linearly from 1.0 to 0.05 over 500×10^3 training steps and kept as constant for the rest of the learning. A mini-batch size of 128 is used for uniformly sampling from relay buffers of 300×10^3 system transitions. The target update frequency is 20×10^3 steps. Latest policy networks after every 10^3 steps are employed to interact with a validation environment 5×10^3 times to compute corresponding cost rates.

The threshold-based maintenance policy used in this comparison study is originated from (Do & Bérenguer, 2012) which can be expressed by a vector $l = [l^1, l^2, \dots, l^N]$ where l^i is the preventive maintenance threshold of component i . The detail description of maintenance schedule of a component i is given in the following. If $s_k^i = m^i$, component i is in failed state. Thus, the "replacement" action is implemented immediately. If $l^i \leq s_k^i < m^i$, component i is still functioning but badly. Therefore, the "imperfect maintenance" action is carried out. If $s_k^i < l^i$, component i is functioning well. Accordingly, the "do nothing" action is chosen.

The optimal system-level preventive maintenance thresholds are obtained via genetic algorithm (GA). During optimization processes, each solution is evaluated using simulation results from 5 runs of 5×10^3 maintenance interventions. The GA optimizer is initialized with a population of 20 elements and a mutation rate of 0.1. The training converges after 25 itera-

tions.

6.3. Simulation results

The simulation results are presented in figure 4, figure 5, table 1 and table 2. It can be noticed that the optimized cost rate of the threshold-based method is highest which is 326.53 (acu) with the corresponding preventive thresholds [1 2 2 2 2 2 2 2 2 2]. The cost rate obtained by BDQ is 265.75 (acu). In comparison with the two others, the customized WQMIX found the best cost rate which is 255.55 (acu).

Based on table 2, it can be seen that the optimization time of the threshold-based policy is shortest due to the fact that the search space of preventive maintenance thresholds is not too large for the studied system. The training time of the customized WQMIX is larger than the one of BDQ because of the extra neural networks used to compute Q and Q^{tot} .

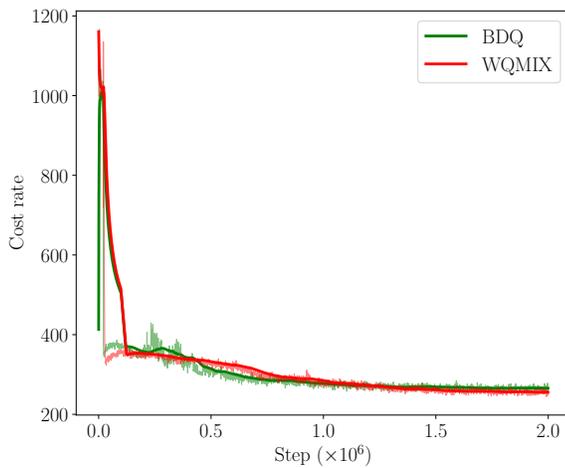


Figure 4. The evolution of cost rates during training

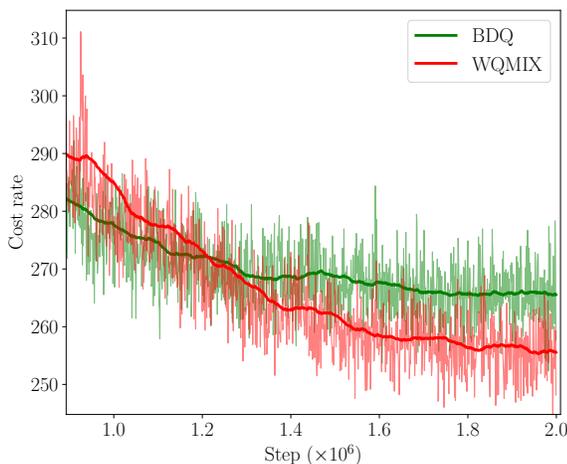


Figure 5. A closer look at the evolution of cost rates

Table 1. Cost rate summary (acu)

WQMIX	255.55
BDQ	265.75
Threshold-based policy	326.53

Table 2. Computing time summary (hours)

WQMIX	12.33
BDQ	8.68
Threshold-based policy	2.35

7. CONCLUSION

In this work, WQMIX algorithm is customized to effectively optimize maintenance decisions of large-scale systems in the case where system states can be fully observable. Particularly, separate agent networks are replaced by a single branching network to take advantage of the fully observable setting. The branching network reserves the ability of avoiding the curse of dimensionality as well as to facilitate decision-making processes. A comparative study is conducted on a 13-component system to examine the performance of the customized algorithm. The obtained results confirmed its effectiveness.

Our future work will focus on CBM modeling approaches for multi-component systems that can integrate multiple kinds of dependencies into maintenance models. Developing MADRL algorithms for maintenance decision optimization will be also considered.

ACKNOWLEDGMENT

This work is part of the AI-PROFICIENT project which has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No 957391.

REFERENCES

- Ahmad, R., & Kamaruddin, S. (2012). An overview of time-based and condition-based maintenance in industrial application. *Computers & industrial engineering*, 63(1), 135–149.
- Andriotis, C., & Papakonstantinou, K. (2019). Managing engineering systems with large state and action spaces through deep reinforcement learning. *Reliability Engineering & System Safety*, 191, 106483.
- Do, P., & Bérenguer, C. (2012). Condition-based maintenance with imperfect preventive repairs for a deteriorating production system. *Quality and Reliability Engineering International*, 28(6), 624–633.
- Ha, D., Dai, A., & Le, Q. V. (2016). Hypernetworks. *ArXiv*.
- Huang, J., Chang, Q., & Arinez, J. (2020). Deep reinforcement learning based preventive maintenance policy for

- serial production lines. *Expert Systems with Applications*, 160, 113701.
- Liu, Y., Chen, Y., & Jiang, T. (2019). Dynamic selective maintenance optimization for multi-state systems over a finite horizon: A deep reinforcement learning approach. *European Journal of Operational Research*, 283(1), 166–181.
- Quatrini, E., Costantino, F., Di Gravio, G., & Patriarca, R. (2020). Condition-based maintenance—an extensive literature review. *Machines*, 8(2), 31.
- Rashid, T., Farquhar, G., Peng, B., & Whiteson, S. (2020). Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*.
- Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., & Whiteson, S. (2018). Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International conference on machine learning* (pp. 4295–4304).
- Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., ... others (2017). Value-decomposition networks for cooperative multi-agent learning. *ArXiv*.
- Tavakoli, A., Pardo, F., & Kormushev, P. (2018). Action branching architectures for deep reinforcement learning. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 32).
- Wang, H. (2002). A survey of maintenance policies of deteriorating systems. *European journal of operational research*, 139(3), 469–489.
- Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., & de Freitas, N. (2016). Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*.
- Wijnmalen, D. J., & Hontelez, J. A. (1997). Coordinated condition-based repair strategies for components of a multi-component maintenance system with discounts. *European Journal of Operational Research*, 98(1), 52–63.
- Zhang, N., & Si, W. (2020). Deep reinforcement learning for condition-based maintenance planning of multi-component systems under dependent competing risks. *Reliability Engineering & System Safety*, 203, 107094.
- Zhou, Y., Li, B., & Lin, T. R. (2021). Maintenance optimisation of multicomponent systems using hierarchical coordinated reinforcement learning. *Reliability Engineering & System Safety*, 108078.

Data Driven Seal Wear Classifications using Acoustic Emissions and Artificial Neural Networks

Nadia. S. Noori¹, Vignesh. V. Shanbhag², Surya. T. Kandukuri³, Rune Schlanbusch⁴

^{1,3} *University of Agder, Department of Engineering Sciences, Jon Lilletuns Vei 9 D, 4879, Grimstad, Norway*

nadia.saad.noori@uia.no
surya.kandukuri@uia.no

^{2,4} *Norwegian Research Centre, Energy & Technology Department, Jon Lilletuns Vei 9 H, 3. etg, 4879, Grimstad, Norway*

vigs@norceresearch.no
rusc@norceresearch.no

ABSTRACT

The work presented in this paper is built on a series of experiments aiming to develop a data-driven and automated method for seal diagnostics using Acoustic Emission (AE) features. Seals in machineries operate in harsh conditions, and seal wear in hydraulic cylinders results in fluid leakage, and instability of the piston rod movement. Therefore, regular inspection of seals is required using automated approaches to improve productivity and to reduce unscheduled maintenance. In this study, we implemented a data-driven diagnostics approach which utilizes AE measurements along with light weight Artificial Neural Networks (ANN) as a classifier to investigate the performance and resources (hardware & software) required for implementing a real-time soft sensor unit for monitoring seal wear condition. We used a feedforward multilayer perceptron ANN (Scaled Conjugate Gradient- SCG algorithm) that is trained with the back propagation algorithm, which is a popular network architecture for a multitude of applications (automotive, oil and gas, electronics). We benchmark the developed method against previous work conducted based on Support Vector Machine (SVM), and we compare ANN performance in classifying the running condition of seals in hydraulic cylinders by applying it to both raw (full frequency spectrum) and down sampled frequency measurements. The experiments were performed at varying pressure conditions on a hydraulic test rig that can simulate fluid leakage conditions like that of hydraulic cylinders. The test cases were generated with seals of three different conditions (unworn, semi-worn, worn). From the AE spectrum, the frequency bands were identified with peak power and by heterodyning the signal. This technique results in 10X down sampling without losing the information of interest. Further,

Nadia. S. Noori et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

the signal was divided into smaller “snapshots” to facilitate rapid diagnosis. In these tests, the diagnosis was made on short-time windows, as low as 0.3 seconds in length. A general set of 16 time and frequency domain features were designed. Then a training set was developed using relevant set of features (4, 5, and 16 features). The data was used to train the ANN (70% training – 30% test & validation) and SVM (60 % training - 40% test and validation). Classification of down sampled measurements, both ANN and SVM were able to accurately classify the status irrespective of the pressure conditions, with an accuracy of ~99% with execution time less than seconds. Therefore, the proposed approach can be applied as part of an automated seal wear classification technique based on AE and ANN/SVM and can be used for real-time monitoring of seal wear in hydraulic cylinders.

Keywords: Hydraulic cylinder, Piston rod seals, Fluid leakage, Acoustic emission, Artificial neural networks.

1. INTRODUCTION

Hydraulic cylinders are linear actuators that are used in a wide variety of material handling applications because they provide high pressure and stable speed rate with low energy consumption (Totten 2011). Stability and efficiency of hydraulic cylinders can be affected due to fluid leakage. Fluid leakage in hydraulic cylinders is mainly due to seal wear, and can affect pressure response, stick-slick of the piston which can further reduce designed efficiency of hydraulic cylinders (Li et al. 2018). Therefore, a maintenance strategy that can identify the seal wear in hydraulic cylinder at initial stages is required. Compared to the reactive and preventive maintenance strategies, condition-based maintenance considers the current health of the system before proposing the maintenance action. Therefore, in this study we propose a condition-based maintenance strategy that can identify the

seal wear and classify the seal wear severity in hydraulic cylinders.

In recent times, attempts have been made to develop robust condition monitoring techniques to monitor seal wear using pressure, vibration, torque, and acoustic emission (AE). For example, (Goharrizi and Sepehri 2011), and (Zhao et al. 2015), monitored the fluid leakage in hydraulic cylinder using pressure sensor and wavelet analysis. (Goharrizi and Sepehri 2011), proposed the root mean square (RMS) feature extracted from level two coefficient to monitor internal leakage and RMS feature extracted from level four coefficient to monitor external leakage, respectively. (Zhao et al. 2015), observed that, displacement signal of the piston rod in hydraulic cylinders was observed to be more sensitive in monitoring hydraulic fluid leakage when compared to pressure signal. In the other work, (Goharrizi and Sepehri 2012), monitored fluid leakage in hydraulic cylinders using pressure sensor and the Hilbert Huang Transform (HHT) technique. The instantaneous magnitude of the first intrinsic mode function (IMF) was proposed as a feature to monitor internal fluid leakage rate. (Petersen et al. 2000) used vibration sensor to monitor seal wear in hydraulic cylinders. The vibration energy (dBVrms) feature was proposed to monitor changes in loading condition and seal wear condition. With increasing loading condition, dBVrms increased, whereas with increasing seal wear condition, dBVrms reduced. (Ramachandran and Siddique 2018) monitored aging of rotary seals using friction torque sensor. Time domain features were extracted from the torque sensor to monitor the seal ageing condition. Features such as mean, RMS, peak and square mean rooted absolute amplitude (SRA) decreased with increase in seal ageing, whereas features such as impulse factor, crest factor and margin factor increased with seal ageing. (Chen, Chua, and Lim 2007), and (Shanbhag et al. 2020) proposed the use of AE to monitor seal wear. (Chen, Chua, and Lim 2007), observed a linear relationship between the RMS feature and the internal fluid leakage rate. Whereas (Shanbhag et al. 2020), proposed the band power and power spectral density (PSD) features to identify fluid leakage due to semi-worn and worn seals. From literature, it can be noted that robust signal-based features have been proposed for seal wear diagnostics.

In recent times, studies have been conducted to diagnose the seal wear condition and classify seal wear severity using signal-based features and data classification or machine learning techniques. For example, (Tang, Wu, and Ma 2010) used wavelet transform with back propagation neural network (BPNN) to analyse the pressure signal and classify different internal leakage severity levels in hydraulic cylinders. Using this method, the authors classified non-leakage, mild leakage, and severe leakage. (Ramachandran and Siddique 2019) used time domain features from the torque signal as an input to the multi-layered perceptron neural network (MLP-NN). Using this technique, an accuracy of 92.86% was achieved in classifying seal wear. In

the other work, (Ramachandran, Keegan, and Siddique 2019), used features from force signal such as maximum force during compression cycle and maximum tension force during the tension cycle to monitor seal degradation. The features from the force signal were used as inputs to the hybrid particle swarm optimization-support vector machine (PSO-SVM) model in classifying seal wear. (Zhang and Chen 2021), proposed AE and the complete ensembled empirical model decomposition (CEEDMAN) technique in classifying different leakage severities such as small, medium, and severe leakage. Using the proposed technique, an accuracy of 93% was observed in classifying different leakage severities. (Kandukuri et al. 2021) used AE and the support vector machine (SVM) for classifying unworn, semi-worn and worn seals. The proposed technique was able to classify the seal wear with an accuracy up to 99%.

From literature we can note that, several attempts have been made to propose diagnostics with machine learning techniques. However, few attempts have been made in studying seal wear classification using artificial neural network (ANN) with AE features as input. Therefore, in this paper, we propose a new methodology in classifying seal wear using AE features and ANN, and we compare the performance of the developed technique with that of the SVM technique. In this paper, the AE data obtained from the tests conducted by (Shanbhag et al. 2021) is used to train, test and validate the ANN model.

2. METHODOLOGY

2.1. Experimental details

The experiments to study seal wear were conducted on a custom-built hydraulic test rig (see Figure 1-a)), that consists of an electro-mechanical cylinder and a hydraulic cylinder head. The hydraulic test rig closely replicates the piston rod and seal interactions, and fluid leakage like that of a hydraulic cylinder. The extension and retraction movement of the piston rod in test rig are driven by a spindle and a nut, that converts rotary motion to translatory motion. A schematic view of the cylinder head (pressurized flange) is shown in Figure 1-b). It consists of three bearing strips to withstand any arising side loads (not present in this study), and piston rod seals acts as fluid sealings. The fluid pressure in the test rig is controlled using pressure relief valves and the pressurized fluid for the cylinder head is supplied through a hydraulic power unit (HPU). A servomotor encoder is used to control the piston rod movement and also record the number of times the piston moves through the cylinder head. At both ends of the extension and retraction strokes there is a dwell time of one second (Stroke length: 600 mm). For the seal wear classification study, only the piston rod seal at the top of the cylinder head (See Figure 1-b)) was replaced with unworn, semi-worn and worn seals (seal material: Polyether-based polyurethane elastomer). Although seal wear in itself is a continuous process, we choose to investigate the

detection capability at specific state of wear. The reason is that the detection process should ideally be economical (computationally cheap) to be realized across multiple units in the industry. It may not be feasible to continuously monitor the signals, only periodic monitoring using AE can classify the seal state and that itself serves as valuable information for maintenance. The unworn seal had no scratches, the semi-worn seal had minor scratches and the worn seal had major scratches on their surfaces (See Figure 2 a-c)). Fluid leakage was observed when semi-worn and worn piston rod seals were used in the cylinder head (See Figure 2 d-e), Fluid type used in test rig: Water glycol. Throughout the experimental study, speed was kept constant at 100 mm/s, and each experiment was conducted for five strokes. For each seal condition experiments were conducted at four pressure conduction: 10, 20, 30, and 40 bar.

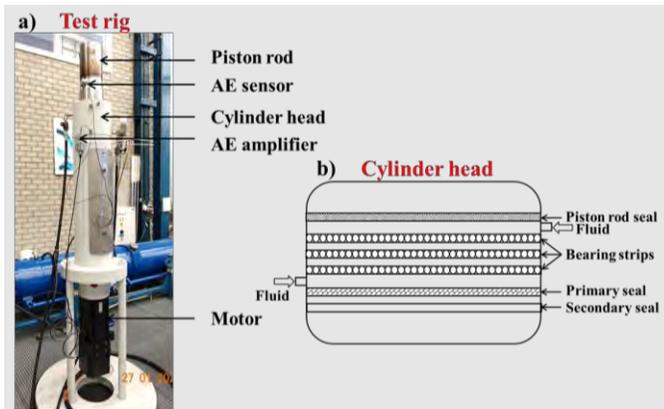


Figure 1. a) Hydraulic test rig, b) Schematic view (view) of cylinder head showing arrangement of piston rod seal and bearing strips.

2.2. AE data acquisition details and bandpass filtering

The AE sensor was mounted on the piston rod for all the experiments as shown in Figure 2-a), as the piston rod is in direct contact with the piston rod seal. The type of AE sensor used in this study was a mid-frequency sensor with resonant frequency at 150 kHz and frequency range of 50-400 kHz. The AE sensor was mounted on the piston rod using an adhesive glue and industrial duct tape to secure a good signal path. The AE sensor was connected to an external pre-amplifier with a gain of 40 dB, and the pre-amplifier was further connected to data acquisition system using co-axial cable. For all the experiments the AE data acquisition was performed at 1 MS/s. In the previous work conducted by (Shanbhag et al. 2021) the AE frequency range that originates from the piston rod seal was observed to be in the frequency range of 50-100 kHz. Therefore, to filter out the frequency range of other parts that are present in the test rig (e.g., spindle, piston rod and bearing strips), heterodyne process was used prior to calculating time and frequency domain features. In the heterodyne process the fast Fourier transform

(FFT) of the signal is calculated and the frequency band of interest (50-100 kHz) is shifted to the origin and remainder of the signal amplitudes in the frequency domain are set to zero (Bechhoefer 2018). A new FFT is calculated, that contains the frequency of interest, but the spectrum is downshifted to 0-50 kHz. Later, the inverse FFT is calculated, and the new signal is down sampled to a new sample rate of 100 kHz, without compromising the information of interest. This technique resulted in reduction of file size by approximately 70%. This reduction in data volume while keeping the frequency content of interest is of significant practical use in monitoring multiple actuators in industrial implementation. The resultant signal is further split into short-time windows as low as 0.28s. From the short time window signals, the time domain and frequency domain features are calculated (Table 1). The implemented technique is beneficial for further research to implement during online monitoring and to perform condition monitoring of parts. Figure 3 represents the methodology adopted for signal processing.

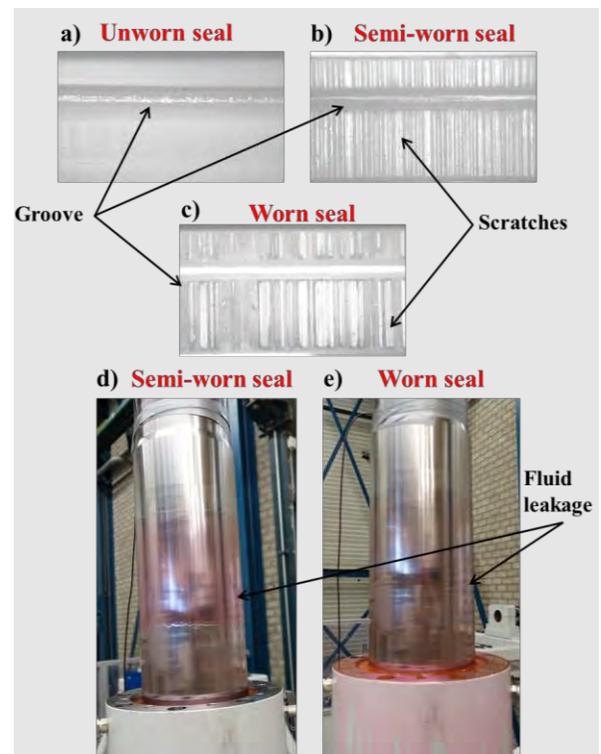


Figure 2. Microscopic camera images of piston rod seals: a) unworn, b) semi-worn, c) worn; fluid leakage on piston rod when d) semi-worn, e) worn piston rod seals were used. (Note: Instrument used take closeup image of a)-c): Jenoptik ProgRes SpeedXT Core 3 CCD Microscope Camera, Pixel size: 3.45 μm X 3.45 μm).

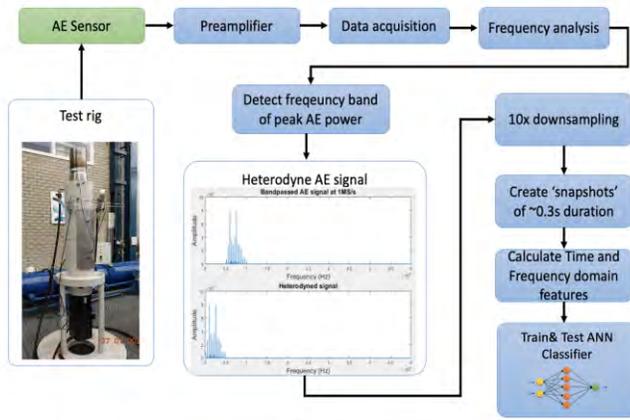


Figure 3. AE signal processing procedure.

Table 1. Time and frequency domain features calculated on AE signal snapshot. (Kandukuri et al. 2021).

Time domain	Frequency domain
$\delta_1 = \frac{\sum_{n=1}^N x(n)}{N}$	$\delta_9 = \frac{\sum_{k=1}^K s(k)}{K}$
$\delta_2 = \sqrt{\frac{\sum_{n=1}^N (x_n - \delta_1)^2}{N}}$	$\delta_{10} = \frac{\sum_{k=1}^K (s(k) - \delta_9)^2}{K}$
$\delta_3 = \sqrt{\frac{\sum_{n=1}^N x(n)^2}{N}}$	$\delta_{11} = \frac{\sum_{k=1}^K (s(k) - \delta_9)^3}{K}$
$\delta_4 = \max(x(n))$	$\delta_{12} = \frac{\sum_{k=1}^K (s(k) - \delta_9)^4}{K}$
$\delta_5 = \frac{\sum_{n=1}^N (x(n) - \delta_1)^3}{\delta_2^3 (N - 1)}$	$\delta_{13} = \frac{\sum_{k=1}^K ks(k)}{\sum_{k=1}^K s(k)}$
$\delta_6 = \frac{\sum_{n=1}^N (x(n) - \delta_1)^4}{\delta_2^4 (N - 1)}$	$\delta_{14} = \sqrt{\frac{\sum_{k=1}^K (k - \delta_{13})^2}{K}}$
$\delta_7 = \frac{\delta_4}{\delta_3}$	$\delta_{15} = \frac{\sum_{k=1}^K (k - \delta_{13})^3 s(k)}{K \delta_{14}^3}$
$\delta_8 = \frac{\delta_4}{\frac{1}{N} \sum_{n=1}^N \sqrt{ x(n) ^2}}$	$\delta_{16} = \frac{\sum_{k=1}^K (k - \delta_{13})^4 s(k)}{K \delta_{14}^4}$
Where $x(n)$ is the signal time series, $n = 1, 2, \dots, N$.	Where $s(k)$ is the frequency spectrum, $k = 1, 2, \dots, K$.

2.3. Artificial Neural Networks (ANN)

The artificial neural networks computing technique is inspired by biological neuron processing. ANN models have been widely applied in various fields of science and technology involving time series forecasting, pattern recognition and process control (Zhang 2003) and (Manoonpong, Pasemann, and Roth 2007). There are multitudes of network types available for ANN applications and its choice depends on the nature of the problem and data availability (Dobrzycki, Mikulski, and Opydo 2019).

¹ Neural Net Fitting, MathWorks documentation, <https://se.mathworks.com/help/deeplearning/ref/neuralnetfitt ing-app.html>

Machine learning methods such as support vector machines (SVM) and ANN or Convolutional NN were used to analyse AE measurements in the domain of structural health monitoring related to wear and tear of equipment (Deshpande, Pandiyan, Meylan, & Wasmer, 2021; Zhao, 2021; Sikdar, Liu, & Kundu, 2022). As the work in this paper is an extension to the research done by (Kandukuri et al. 2021) and (Noori et al. 2020) for addressing the need for reliable and accurate non-intrusive measurements for seal condition. The methods presented here may be used in many sectors satisfying all these requirements, yet some changes in these expected features may be seen in the practices of different seals, and changes might be necessary to adapt the current proposed method.

In this research, a feedforward multilayer perceptron ANN was utilized, that was trained with the backpropagation algorithm. In this type of network, the artificial neurons, or processing units, are arranged in a layered configuration containing an input layer, usually one “hidden” layer, and an output layer. Units in the input layer introduce normalized or filtered values of each input into the network. Units in the hidden and output layers are connected to all the units in the preceding layer. Each connection carries a weight factor. The weighted sum of all inputs to a processing unit is calculated and compared to a threshold value. Then activation signal is then passed through a mathematical transfer function to create an output signal, that is further sent to the processing units in the next layer. In this study, we used MATLAB – neural net fitting tool (NFTOOL) to carry out the data analysis and generating the ANN models (see Figure 4). The toolbox offers three options of shallow neural networks: Levenberg-Marquardt, Bayesian regularization, and Scaled Conjugate Gradient backpropagation. Due to the large size of the dataset, we used the Scaled Conjugate Gradient backpropagation algorithm (Møller, 1993). As it is also recommended in the NFTOOL documentation: “Scaled conjugate gradient backpropagation updates weight and bias values according to the scaled conjugate gradient method. For large problems, scaled conjugate gradient is recommended as it uses gradient calculations which are more memory efficient than the Jacobian calculations used by Levenberg-Marquardt or Bayesian regularization.”¹. The SCG - ANN architecture used consists of one hidden layer with five or 10 neurons.

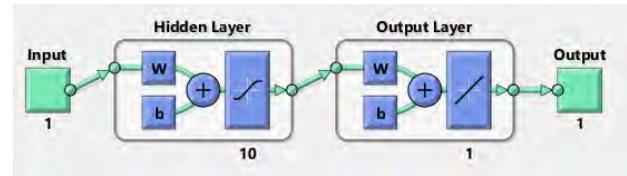


Figure 4. ANN Architecture using MATLAB NFTOOL.

2.4. ANN model training and evaluation

In this study, we prepared three datasets to carry out the analysis and compare the inputs and SCG ANN models that would generate optimal results. The first dataset represents *filtered raw data* time series with 4096 X 63250 points of measurements. Second dataset represents *features filtered* time series with 13 X 63250 points of measurements. The third dataset represents *features filtered downsized* with 16 X 1200 points of measurements. The datasets were used for training, validating, and testing three ANN models created using the NFTOOL (See Figure 4) and used Scaled Conjugate Gradient algorithm to carry out the tests. Data from the AE sensor measurement, i.e., filtered frequency bands (4096 inputs), features filtered (raw 13 inputs or downsized 16 inputs) were used as input vector for the model. As for the output vector, the seal condition vector for every measurement was constructed manually to describe three operational conditions: worn (1), semi-worn (2) and unworn (3) (See Figure 5 for example of signal signature of the three seal conditions). The seal condition vector was set as the target vector for all the SCG ANN models' training. The dataset samples were divided into three sets used for training (70%), validation (15%) and testing (15%).

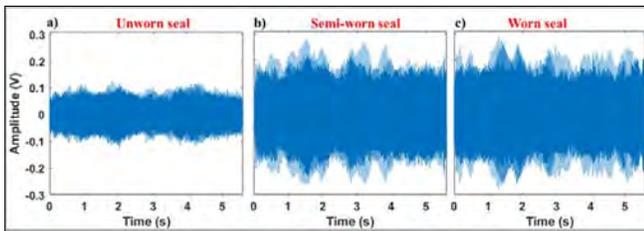


Figure 5. Signal signature samples for the seal conditions: unworn (3), semi-worn (2), and worn (1).

3. RESULT AND DISCUSSION

The results for the analysis showed variation in classification accuracy when using different variable types or different number of variables as input to the ANN models. In addition, to implementing changes to the input vector of neural network model, we also implemented changes in the number of neurons to reach optimal results. The results for the different combinations of datasets and neural network models are listed in Table 2. Figure 6 shows the confusion matrix results for the used cases listed in Table 2. In Figure 7, we provide details related to the setup of the ANN model used in case #5, the ANN option used is SCG backpropagation options in NFTOOLS for the experiments. The NFTOOL provided us with different key performance indicators to learn about the model performance in relation to the classification accuracy, needed time for training, and other information, that would help understanding the model behaviour too. In Figure 7, it shows the number of training epochs was 128 with ~0 second time for the training. The network used has 16 inputs, a hidden layer with 5 nodes and

three outputs. Worth to mention, the time for training the SCG model in Case #1 using the raw data with 4046 inputs, was 3 minutes and 37 seconds with 110 epochs to finish the training. The training time using the downsized data, was ~0 sec for all cases.

We can observe from the results that using the ANN solution to classify condition of the seal is possible with a high accuracy rate, using all features (i.e., case #1, case #3, case #5) for filtered and downsized data. Furthermore, we can observe that in the case of using a subset (i.e., case #2, case #4, case #6) of the features from the downsized dataset, the ANN model was able to provide a high accuracy seal condition classification result. The experiments related to model tuning, we tried different options related to the number of inputs (4069, 13, 4, 5, 10), and number of hidden nodes (5 or 10 nodes). When used, there was no great improvement in the classification results. As for the number of inputs for the network, we experimented with different combinations of input features (i.e., raw data 4069, all 16 features, sub-set of features 5 features) to examine the impact of number of features' selection on the improvement of the classification results. As the results showed, that using part of the features set can yield good results. However, we have not experimented with different combinations to find the most representative set of features that can be used instead of the whole dataset, to optimize the solution further.

Furthermore, the shallow ANN is a lightweight neural network architecture, providing us with the ability to use the full dataset for the filtered data before (i.e., case #1 and case #2) and after downsizing, with a classification accuracy rate varying between 0.8-0.99. For training the ANN, a normal laptop was used (8 GB RAM, Intel Core i5 CPU, no GPU), and the training times were approximately 0 seconds, except in the raw data Case #1. Thus, from automation and deployment of the method, shallow ANN is promising as it does not require specialized equipment to train the network and to later deploy it.

However, we can see, when we reduce the number of chosen features to five, and only use five neurons for the hidden layer in SCG-ANN model (using the options offered in NFTOOL setup wizard), the classification accuracy is impacted drastically, as the accuracy was reduced to 0.692. However, with 16 features, the network with five neurons provided the best accuracy classification results.

Hence to compare the developed method against previous work conducted using Support Vector Machine (SVM) (Kandukuri et al,2021), the ANN performance in classifying the seal condition by analysing both raw (full frequency spectrum) and down sampled frequency measurements were considered. The SVM results showed that worn seal condition was classified accurately under all the conditions, whereas accuracy of 99.4 % and 98.1 % were observed for the unworn and semi-worn cases, respectively. Compared to the test cases experiments in this paper, it can be observed

that best case is Case #5, with overall classification accuracy of 99.3, and classification accuracy rate of 99.9% for unworn, 100% for semi-worn, and 98.8% for worn. As mentioned in section 1, Zhang and Chen (2021), proposed AE and the complete ensembled empirical model decomposition (CEEDMAN) technique in classifying different leakage severities such as small, medium, and severe leakage. However, the proposed technique, provided an accuracy of 93% in classifying different leakage severities.

In general, the classification of down sampled measurements, both ANN and SVM were able to accurately classify the status irrespective of the pressure conditions, with an accuracy of ~99% - 100% with execution time less than seconds. Therefore, the proposed approach can be applied as part of an automated seal wear classification technique based on AE and ANN/SVM and can be used for real-time monitoring of seal wear in hydraulic cylinders.

The Shallow ANN advantage over the SVM approach that was followed in our previous study (Kandukuri et al. 2021) is the flexibility in configuring the ANN rapidly to improve classification accuracy results, with less required resources.

Table 2. Summary of the results for classifying the seal condition using AE sensor data, full samples and downsized.

Case #	Input and # of neurons	Accuracy
#1	<ul style="list-style-type: none"> Filtered data – time series – 4096 X 63250 Using 4096 inputs Network size (10 hidden, 3 output) 	0.81
#2	<ul style="list-style-type: none"> Features filtered – 13 X 63250 Using 4 features, namely (δ_6, δ_{10}, δ_{11}, and δ_{12}) / 4 inputs Network size (10 hidden, 3 output) 	0.867
#3	<ul style="list-style-type: none"> Features filtered downsized – 16 X 1200 Using all time and frequency domains 16 features / 16 input Network size (10 hidden, 3 output) 	0.856
#4	<ul style="list-style-type: none"> Features filtered downsized – 16 X 1200 Using 5 features, namely (δ_5, δ_6, δ_{12}, δ_{14}, and δ_{15}) / 5 input. Network size (10 hidden, 3 output) 	0.802
#5	<ul style="list-style-type: none"> Features filtered downsized – 16 X 1200 	0.993

	<ul style="list-style-type: none"> Using 16 features/16 inputs Network size (5 hidden, 3 output) 	
#6	<ul style="list-style-type: none"> Features filtered downsized – 16 X 1200 Using only 5 features/5 inputs, namely, (δ_5, δ_6, δ_{12}, δ_{14}, and δ_{15}) Network size (5 hidden, 3 output) 	0.62



Figure 6. Confusion matrices for the results presented in Table 3. Seal condition codes: unworn (3), semi-worn (2), and worn (1).

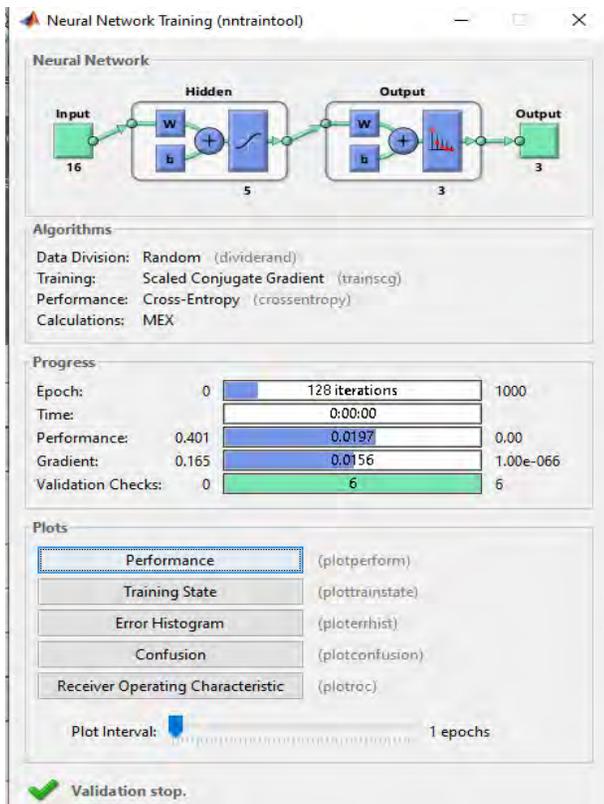


Figure 7. Details of ANN training and ANN architecture for case #5.

4. SUMMARY

In this work we have presented an approach to apply a data-driven solution in combination with using shallow ANN to analyse AE sensor data. The goal was to classify accurately and in real-time the seal condition in a hydraulic cylinder to detect possible leakage. The work was based on a series of experiments carried out on customized test rig, in combination with a data pre-processing pipeline to reduce the size and the variable dimensions of the collected data, then applying machine learning methods for the analysis. In this part we successfully applied an ANN and reached high score of classification accuracy with negligent time for processing input variables with different variants (i.e., raw data to downsized filtered data). The resulted ANN models are easy to deploy and implement as a soft sensor on an edge device such as a Raspberry PI or Jetson Nano, attached to the monitored unit. The advantage is that data will be processed locally and only useful information about the seal condition will be transmitted over wireless or wired connection to the main or respective control system.

ACKNOWLEDGEMENT

The research presented in this paper has received funding from the Norwegian Research Council, SFI Offshore Mechatronics, project number 237896.

REFERENCES

- Bechhoefer, Eric. 2018 “A quick introduction to bearing envelope analysis”. pp 1-10.
- Chen, P., P.S.K. Chua, and G.H. Lim. 2007. “A Study of Hydraulic Seal Integrity.” *Mechanical Systems and Signal Processing* 21(2): 1115–26.
- Deshpande, P., Pandiyan, V., Meylan, B., & Wasmer, K. 2021. "Acoustic emission and machine learning based classification of wear generated using a pin-on-disc tribometer equipped with a digital holographic microscope". *Wear*, 476, 203622.
- Dobrzycki, Arkadiusz, Stanisław Mikulski, and Władysław Opydo. 2019. “Using ANN and SVM for the Detection of Acoustic Emission Signals Accompanying Epoxy Resin Electrical Treeing.” *Applied Sciences* 9(8): 1523.
- Goharrizi, Amin Yazdanpanah, and Nariman Sepehri. 2011. “A Wavelet-Based Approach for External Leakage Detection and Isolation From Internal Leakage in Valve-Controlled Hydraulic Actuators.” *IEEE Transactions on Industrial Electronics* 58(9): 4374–84.
- Goharrizi, Amin Yazdanpanah, and Nariman Sepehri. 2012. “Internal Leakage Detection in Hydraulic Actuators Using Empirical Mode Decomposition and Hilbert Spectrum.” *IEEE Transactions on Instrumentation and Measurement* 61(2): 368–78.
- Kandukuri, Surya T. et al. 2021. “Automated and Rapid Seal Wear Classification Based on Acoustic Emission and Support Vector Machine.” *PHM Society European Conference* 6(1): 8–8.
- Li, Lin et al. 2018. “Featured Temporal Segmentation Method and AdaBoost-BP Detector for Internal Leakage Evaluation of a Hydraulic Cylinder.” *Measurement* 130: 279–89.
- Manoonpong, Poramate, Frank Pasemann, and Hubert Roth. 2007. “Modular Reactive Neurocontrol for Biologically Inspired Walking Machines.” *The International Journal of Robotics Research* 26(3): 301–31.
- Møller, M. F. 1993. A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, 6(4), 525-533.
- Noori, N. S. et al. 2020. “Non-Newtonian Fluid Flow Measurement in Open Venturi Channel Using

- Shallow Neural Network Time Series and Non-Contact Level Measurement Radar Sensors.” In *Day 1 Mon, November 02, 2020*, Virtual: SPE, D011S001R003.
<https://onepetro.org/SPEBERG/proceedings/20BERG/1-20BERG/Virtual/448656> (April 12, 2022).
- Petersen, Dr et al. 2000. “Condition Monitoring of a Water Hydraulic Cylinder by Vibration Analysis.” *Journal of Testing and Evaluation* 28(6): 507.
- Ramachandran, Madhumitha, Jon Keegan, and Zahed Siddique. 2019. “Hybrid PSO-SVM Based Method for Degradation Process Prediction of Reciprocating Seal.” *Annual Conference of the PHM Society* 11(1).
<https://papers.phmsociety.org/index.php/phmconf/article/view/852> (April 12, 2022).
- Ramachandran, Madhumitha, and Zahed Siddique. 2019. “A Data-Driven, Statistical Feature-Based, Neural Network Method for Rotary Seal Prognostics.” *Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems* 2(2): 024501.
- Shanbhag, Vignesh V., Thomas J. J. Meyer, Leo W. Caspers, and Rune Schlanbusch. 2020. “Condition Monitoring of Hydraulic Cylinder Seals Using Acoustic Emissions.” *The International Journal of Advanced Manufacturing Technology* 109(5–6): 1727–39.
- Shanbhag, V.V., Meyer, T.J.J., Caspers, L.W., Schlanbusch, R. 2021. “Defining Acoustic Emission-Based Condition Monitoring Indicators for Monitoring Piston Rod Seal and Bearing Wear in Hydraulic Cylinders.” *The International Journal of Advanced Manufacturing Technology* 115(9–10): 2729–46.
- Sikdar, S., Liu, D., & Kundu, A. 2022. Acoustic emission data based deep learning approach for classification and detection of damage-sources in a composite panel. *Composites Part B: Engineering*, 228, 109450.
- Tang, Hong Bin, Yun Xin Wu, and Chang Xun Ma. 2010. “Inner Leakage Fault Diagnosis of Hydraulic Cylinder Using Wavelet Energy.” *Advanced Materials Research* 139–141: 2517–21.
- Totten, George E. 2011. *Handbook of Hydraulic Fluid Technology*. CRC Press.
- Zhang, G.Peter. 2003. “Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model.” *Neurocomputing* 50: 159–75.
- Zhang, Peng, and Xinyuan Chen. 2021. “Internal Leakage Diagnosis of a Hydraulic Cylinder Based on Optimization DBN Using the CEEMDAN Technique” ed. Li Qing. *Shock and Vibration 2021*: 1–10.
- Zhao, K. 2021. A Literature Review on the Application of Acoustic Emission to Machine Condition Monitoring. [Master's thesis in Smart Energy - University of Vaasa] School of Technology and Innovations, university of Vaasa, Finland.
- Zhao, Xiuxu et al. 2015. “Experimental Study of Hydraulic Cylinder Leakage and Fault Feature Extraction Based on Wavelet Packet Analysis.” *Computers & Fluids* 106: 33–40.

Severity Estimation of Faulty Bearings Based on Strain Signals From Physical Models and FBG Measurements

Ravit Ohana¹, Renata Klein², Jacob Bortman¹

¹*PHM Laboratory, Department of Mechanical Engineering, Ben-Gurion University of the Negev, P.O.B 653, Beer-Sheva 8410501, Israel*

*ravitoh@post.bgu.ac.il
jacbort@gmail.com*

²*R.K. Diagnostics, P.O.B 101, Gilon, D.N. Misgav 20103, Israel*

Renata.Klein@RKDiagnostics.co.il

ABSTRACT

Condition based maintenance (CBM) is the preferred approach in rotating machinery and aim to replace the commonly used approach of maintenance based on service time. To achieve an effective CBM, different types of sensors should be placed in the system for condition monitoring to detect the location of the fault and its severity. In this research, a Fiber Bragg Grating (FBG) has been used for condition monitoring on spalls in deep groove ball bearings. The motivation for using these sensors is the ability to get a high-noise signal (SNR) ratio. The usage of FBG sensors is relatively new for health monitoring systems of rotating machinery. Therefore, there is not enough understanding of the strain signature measured by the FBG. To examine the phenomena in the strain signals, a physics-based model of the strain signature has been developed. In this model, two complementary models were integrated, a finite element (FE) model and a dynamic model. The strain model describes the interaction between the rolling elements (REs) and the bearing housing and simulates the strain behavior measured on the bearing housing. The simulation results are validated with strain signals measured by the FBG sensor at different stages of an endurance test. The model allows simulation of a wide range of spall lengths and describes the behavior of the strain signals for different levels of misalignment. The insights from the model enabled the development of an automatic algorithm that assess the severity of the

defect and to track spall length during bearing operation, based on strain signals.

1. INTRODUCTION

Bearings are important components in rotating machinery to enable smooth operation. Failure in one of the bearings can lead to severe damage and even total failure. The most common failure mechanism in bearings is spall formation (Jalalhmadi *et al.*, 2009), where metallic particle flakes are released from the surface of the bearing raceways and/or the rolling elements (REs) (Raje, Slack and Sadeghi, 2009; Rosado *et al.*, 2010; Gazizulin, Klein and Bortman, 2017). Interactions with a defective component within a bearing lead to undesirable vibrations which increase as the defect evolves. Fault detection and diagnosis, based on vibration signals (such as, accelerometers, load cell, acoustic emission etc.), have been widely studied in the literature (Qing *et al.*, 1991; Zhen *et al.*, 2008; Eftekharnjad *et al.*, 2010; Randall and Rôme Antoni, 2010; Randall, 2011; Cui *et al.*, 2015; El-Thalji and Jantunen, 2015; French and Hannon, 2015; W.S. Siew, W.A. Smith, Z. Peng, 2015; Wang, Sawalhi and Becker, 2016; Tarawneh *et al.*, 2019; Zhang *et al.*, 2021, 2022). This study examines strain signals measured with Fiber Bragg Grating (FBG). FBG are small, flexible and gives a high signal to noise ratio (SNR).

Fault diagnosis of bearings via FBG sensors has not been thoroughly studied in the literature (Khmelnitsky *et al.*, 2015; Wei *et al.*, 2016; Alian *et al.*, 2019). Numerous researchers have developed physics-based models to investigate the RE-edge impact reaction (Arakere *et al.*, 2009; Branch *et al.*, 2009, 2013;

First Author et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Gazizulin *et al.*, 2020). Branch *et al.* (Branch *et al.*, 2013) presents a FE analysis of RE-spall impact to determine the relative contributions of four bearings' properties on spall propagation: (i) ball material density, (ii) subsurface residual stress, (iii) gradient in yield strength with depth (case hardening), and (iv) raceway surface hardness/yield strength that are thought to affect spall propagation. None of these studies investigated the strain states in a remote area from the spall. A fundamental study to examine the physical phenomenon in the strain signal, during the interaction of the RE with the damaged part of the bearing, is presented. In this study, a strain model has been developed and compared to the experimental strain signal measured by FBG sensors. This article deals with small spalls, such that the RE meets the spall edge before it collides with the floor of the spall.

The paper is organized as follows. Section 2 presents the theoretical background. Section 3 presents the experimental setup. Section 4 presents the strain model. Section 5 presents a comparison between simulations and experimental results and Section 6 presents the algorithm for spall length estimation.

2. BACKGROUND

Strain signal has lack of interpretation in the literature. For this purpose, two complementary models were integrated into a strain model: a FE model and a dynamic model. A strain-contact force relation was developed using the FE model. This relation is multiplied by the contact force from the dynamic model to simulate the real strain behavior of the bearing house. The simulated strain and the strain signals measured by the FBG in the endurance test were compared to validate the model and to explain different phenomena detected in the strain signals. The flowchart methodology presented in Figure 1. Based on the strain model insight, a spall size estimation algorithm was developed.

2.1. FBG Signal

A FBG sensor is made by the inscription of periodic variations in the index of refraction along a short section of the core of a single-mode optical fiber (Hill and Meltz, 1997). The strain signals measured by the FBG sensor, given by

$$\varepsilon = \frac{\lambda_b - \lambda_0}{C_\varepsilon \lambda_0} \quad (1)$$

Where λ_b is the light reflection in a very narrowband of wavelengths around the Bragg wavelength of peak reflection, λ_0 and C_ε is a material constant.

The usage of FBG sensors is motivated by the ability to install them close to the tested bearing, which gives a high signal to noise ratio (SNR).

2.2. Data Analysis

The strain signal in the time domain was synchronized with the rotational speed signal, by angular resampling, and was transformed to the cycle domain. A de-phase algorithm (Klein *et al.*, 2011) was applied to the strain in the cycle domain to remove the synchronous elements, e.g. shaft, that were masking the effect of the bearing in the signal. First, the synchronous average (SA) is calculated by averaging the resampled signal over a cycle of rotation. The de-phased signal is obtained by removing the SA from the angular resampled signal, leaving only the asynchronous elements. The data analysis process of the strain signal is presented in Figure 2.

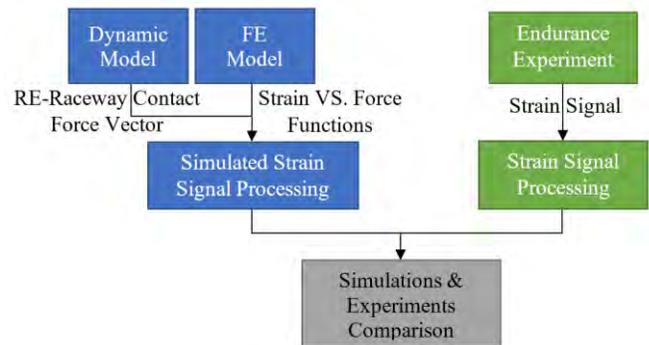


Figure 1: Research flowchart structure

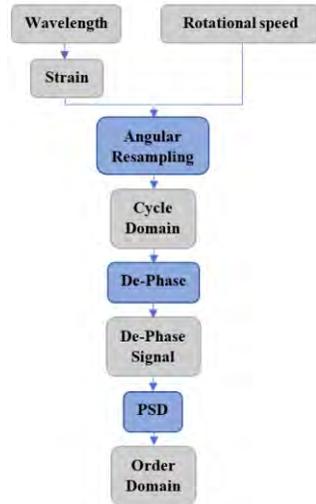


Figure 2: Strain signal analysis process

3. ENDURANCE TEST

The experimental setup for the endurance test consists of a test rig and a measurement unit. The test rig includes a shaft, two support bearings, the tested bearing (6206 ETN9) and a pneumatic piston for applying different loads. All components are listed in Table 1 and marked in Figure 3. All the bearings were lubricated. A FBG sensor mounted along the bearing house of the tested bearing, see Figure 3. The sensor is connected to a Smart Fiber™ “SmartScan Aero Mini” interrogator, sampling data at 10[kS/s]. To accelerate the defect initiation process, a small bore was seeded into the outer raceway of the bearing using an electrical discharge machining (EDM). The initial defect was in the center of the loading zone. The operational conditions during the test were constant with a load of 2.2 kN and the rotation speed was 35 Hz.

Figure 4 presents the strain de-phased signal at three different stages of the experiment: green signal-at the beginning of the experiment, the blue signal-at the middle of the experiment and the red signal-at the end of the experiment. All three of the signals, have a form of a periodic signal wave with BPFO frequency. An additional pulse was observed at the advanced stages of the experiment. In addition, the pulse widths and amplitude levels are higher for advanced stages of the experiment with wider spalls.

Table 1: Experimental system components

Component number	Component	Amount
1,2	Supporting bearing SYF 30 TF	2
3	Coupling	1
4	Shaft 30 mm diameter	1
5	Bearing house + tested bearing 6206 ETN9	1
6	Gear 20 teeth 2.5 module	1
7	Pneumatic piston	1
8	Three-phase motor	1

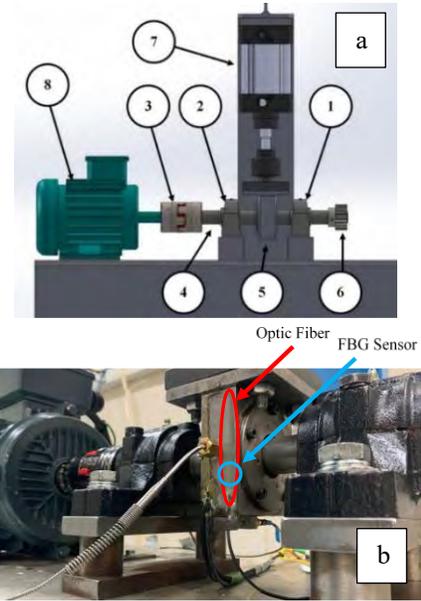


Figure 3: Endurance test rig: (a) Illustration of the experimental system. The component details are listed in Table 1. (b) Location of the FBG sensor.

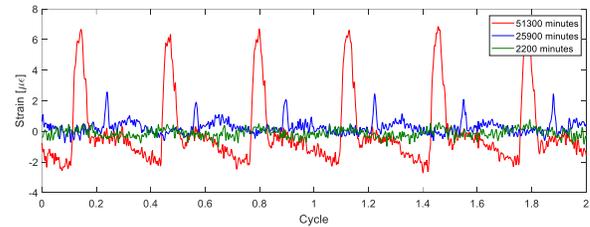


Figure 4: Strain de-phased signal at different stages during the endurance experiment.

4. STRAIN SIGNAL MODEL

To analyze the behavior of the measured FBG strain signals for healthy and faulty bearings, the strain model integrates between a dynamic model and a FE model as illustrated in the research flowchart diagram presented in Figure 1.

4.1. Non-Linear Dynamic Model

A validated non-linear dynamic model was used to calculate the contact force between the RE and the outer raceway generated during the operation of both a healthy and a faulty bearing. The non-linear dynamic model developed by Kogan et al. (Kogan *et al.*, 2015; Kogan, Bortman and Klein, 2017) is based on the classic kinematics and the dynamic equations. The boundary conditions for the dynamic model were defined according to the endurance experimental setup. Two different spall lengths were examined: 2mm and 4.8 mm. An example of the contact force of a faulty

bearings with a spall length of 2mm versus the rotation angle, θ , (as illustrated in Figure 6a), of one RE is presented in Figure 6b. In Figure 6d, the green dot, and the red dot in the $\theta=270^\circ$ region, represents the entry and exit of the RE into and from the spall, respectively. The high frequency appears in the signal is a result of the RE-spall interaction at the trailing edge. The zero force in this region represent the RE disconnection from the raceways (destressing), i.e., free flight of the RE.

4.2. FE Model

A FE model was built to study the relation between (i) the contact force between the REs and the outer raceway, and (ii) the strain developed within the bearing house. The strain response location was chosen to be the same as the FBG sensor located in the test rig. A quasi-static, two-dimensional, plane strain model was developed in ABAQUS software with the ABAQUS/standard solver. The assembly included a single RE and a bearing house (Figure 5a). The FE model geometry and dimensions presented in Figure 5a and in Table 2. The Young's modulus was taken as $E = 208 \text{ MPa}$ and the Poisson ratio was $\nu = 0.3$. An eight-node biquadratic element was used (CPE8R). A fine mesh was used in the vicinity of the contact zone and became coarser as the distance from the contact zone increased. The narrow edge was fixed at the top surface (see Figure 5a). The RE was pressed, with constant force, against the bearing house. Then, the RE was rotated 360 degrees counterclockwise with angle increments of 0.005 radians ($\Delta\theta = 0.005 \text{ rad}$) until it returned to the starting point. The simulated strain from the FE model for constant force is shown in Figure 5b. Since the contact force between the RE and the outer raceway is changing during the rotation of the RE, i.e., inside, and outside of the loading zone, the integration of the dynamic model is essential. A linear correlation between the FE strain and the constant force acting on the RE, was sought regarding the fact that, the measurement of the strain was at a far distance from the contact zone and the material is linear elastic. It is noteworthy to mention that the linear function was calculated for every angle increment.

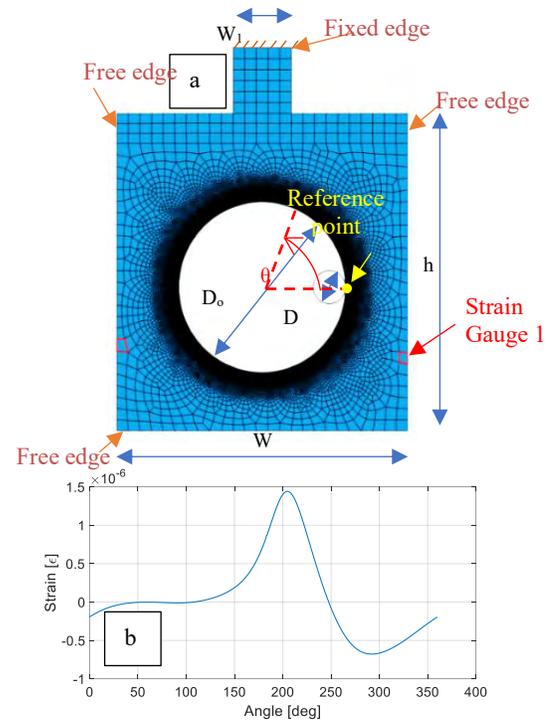


Figure 5: (a) FE model – dimensions. (b) Simulated strain results obtained from the FE model after applying a constant force of $F=100 \text{ [N]}$ on the RE

Table 2: Model dimensions

Parameter	Comment	Value	Units
h	Bearing house height	106	mm
w	Bearing house width	100	mm
d	Bearing house depth	32	mm
D	RE diameter	11	mm
D_o	Outer race diameter	57	mm
w_1	Piston axis diameter	20	mm
N	RE number	8	-

4.3. Simulated Strain Signal

The simulated strain signal has obtained by multiplying the contact force from the dynamic model (Section 4.1) and the slope of the linear functions from the FE model (Section 4.2) for every angle increment. An example of the simulated strain of faulty bearing with 2mm spall length is shown in Figure 6c and e. The combined effect of (i) the behavior of the contact force distribution (Figure 6b), and (ii) the behavior of the simulated strain from the FE model (Figure 5b) is observed. The

calculated strain in the faulty state (Figure 6e) has the same shape as the contact force signal in the RE-spall interaction (Figure 6d), which means that the disconnection of the RE from the raceways can also be observed in the strain signals.

5. RESULTS

The model was validated by comparing the simulation results, Figure 7a, to the experiments, Figure 7b. The green curve in both graphs represent a healthy bearing and the red curve represent a faulty bearing with 4.8mm spall length. There is good agreement between the

results of the simulation and the experiments, both in behavior and in scale. The simulated signal is biased, because the model simulates the total strain including the loading effect while the FBG sensor measures only the strain fluctuations. As illustrated in Figure 8, when a single RE reach to the spall leading edge it may impact different locations with the trailing edge. In this Figure, two examples of different impact locations are shown. For each one of the impact a different pulse width will be measured. Although the pulse width represents very well the hovering of the RE, it can give a good estimation of the spall length.

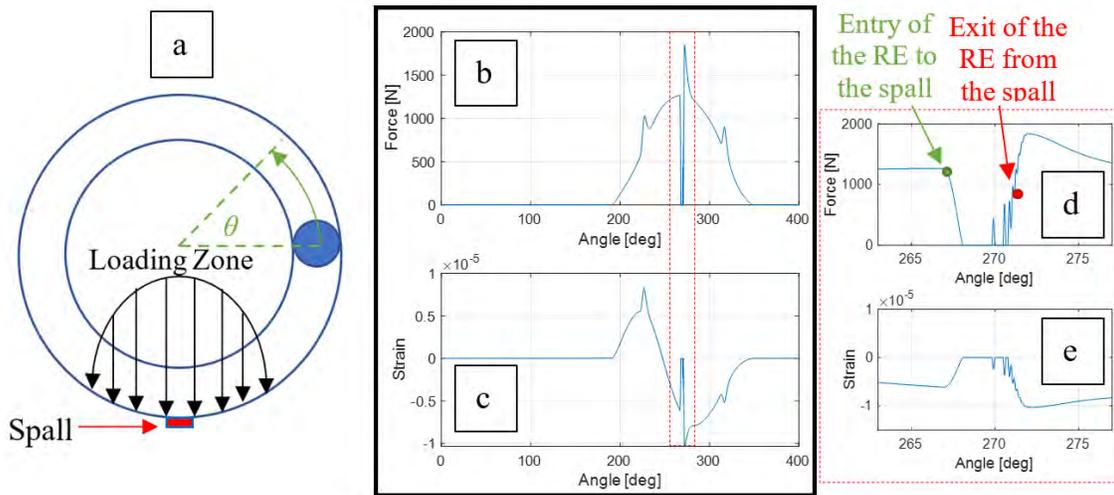


Figure 6: (a) Schematic description of the bearing from the endurance test: the loading zone is depicted by the black curve and the location of the spall can be seen in red rectangle. The angle of rotation, θ , relative to the reference point (b) contact force of faulty bearing with 2mm spall length, and (c) simulated strain signal in strain gauge 1 (Figure 5a) for faulty bearing (2mm spall), (d) and (e) close-up of the RE-spall interaction of the contact force and the simulated strain signal respectively

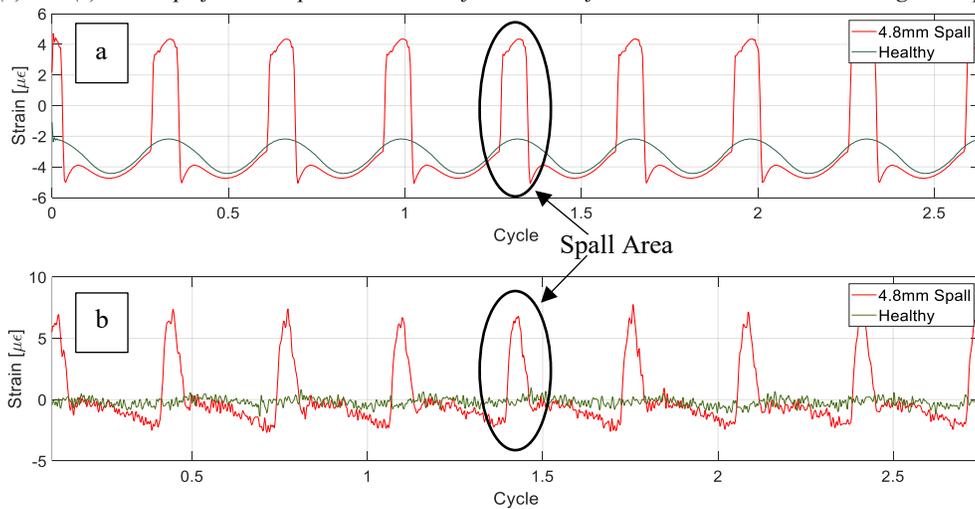


Figure 7: Comparison between: (a) simulation and (b) experimental endurance test (healthy - green and 4.8mm spall - red)

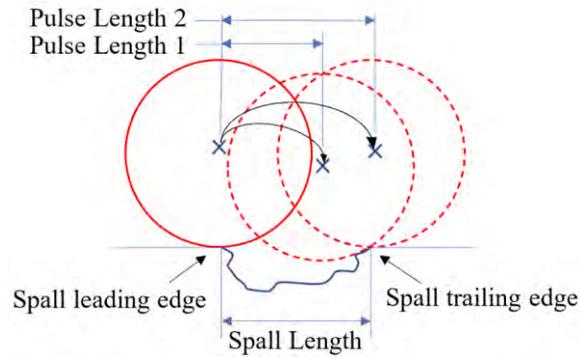


Figure 8: Illustration of a RE impact in two different possible locations at the spall edge in the experiments

6. SPALL LENGTH ESTIMATION ALGORITHM DESCRIPTION

The spall length estimation algorithm is divided into two main stages and presented in Figure 9: (i) selection of the RE-spall interactions and (ii) evaluation of the pulse width that represents the RE flight, based on the Defect Size Ratio (DSR) equation (Alian *et al.*, 2019):

$$DSR = \frac{\Delta x}{X} = \frac{\Delta C}{C} \quad (2)$$

Where Δx is the spall length, X is the distance between two RE, ΔC is the pulse width in the cycle domain and C is the distance between two pulses in the cycle domain.

The selection of the RE-spall interactions is performed on the squared de-phased strain signal, x^2 , in order to emphasize the pulses in the strain signal. Since the spall is located on the outer raceway of the bearing, it is possible to detect BPFO interactions in one cycle. A dynamic threshold, which is a percentage of the highest value in the signal, is used to determine which peaks represent the correct interactions. After the RE-spall interaction has been detected, the algorithm searches the entrance and the exit points of the RE into the spall by defining the first negative minimum adjacent to the peak of the pulse. The evaluated pulse width, that it is assumed to be the spall length, is obtained by the average of all the estimated pulse widths.

The pulses started to be observed in the signals after 24,000 minutes, therefore, the algorithm was applied to all the strain de-phased signals after 24,000 minutes. Figure 10 presents the algorithm output which is a trend line of the estimated spall length. The trend line is approximately monotonic as expected. The minimal spall length that could be estimated by the algorithm was 1.6 mm. After the bearing was disassembled, a 5.1 mm spall was measured, while the estimated fault

length was 5.4 mm. The approximation error is close to 6%.

This work focuses on outer ring defects. For further research, a general algorithm (both the outer and inner rings defects and rolling elements defects) will enable to develop an industrial application.

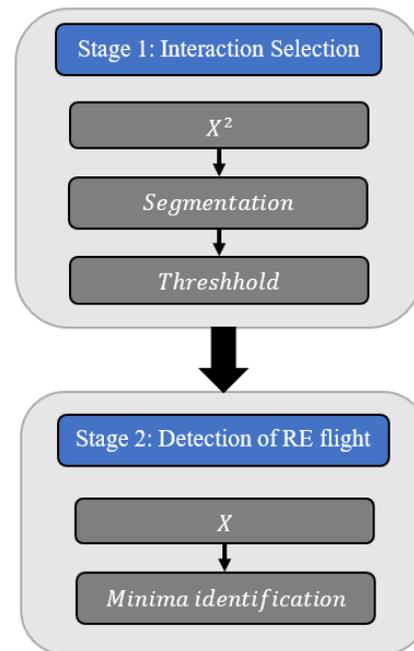


Figure 9: Algorithm stages for spall length estimation in the bearing outer race for the strain signal

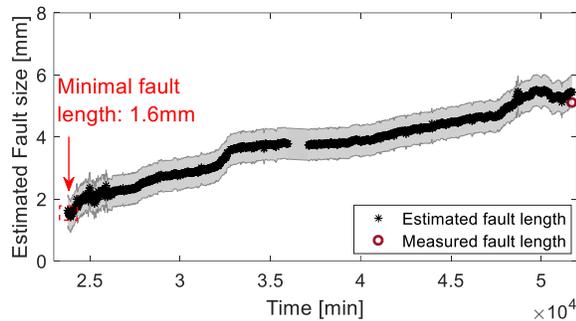


Figure 10: Trend line of the fault length estimation during the endurance experiment

7. SUMMARY AND CONCLUSIONS

This study examines the strain signals of healthy and faulty bearings, for diagnostic capabilities via FBG sensors. The theoretical analysis is based on two complementary models that were integrated into a strain model. The validation of the model has been done by comparing the simulation results with the endurance test results. Based on the model insights, it was possible to determine that the additional pulses correspond to the free flight of the ball over the spalled area. In addition, the pulse width is directly related to the arc length of the free flight of the ball. Based on the strain model, a reliable algorithm for spall length estimation has been developed.

ACKNOWLEDGEMENT

I gratefully express my deepest appreciation to the Pearlstone Center for their support and funding of this work.

REFERENCES

- Alian, H. *et al.* (2019) ‘Bearing fault detection and fault size estimation using fiber-optic sensors’, *Mechanical Systems and Signal Processing*, 120, pp. 392–407.
- Arakere, N. K. *et al.* (2009) ‘Rolling contact fatigue life and spall propagation of AISI M50, M50NiL, and AISI 52100, part II: Stress modeling’, *Tribology Transactions*, 53(1), pp. 42–51.
- Branch, N. *et al.* (2009) ‘Stress field evolution in a ball bearing raceway fatigue spall’, *Journal of ASTM international*, 7, pp. 1–18.
- Branch, N. A. *et al.* (2013) ‘Critical stresses and strains at the spall edge of a case hardened bearing due to ball impact’, *International Journal of Fatigue*, 47,

pp. 268–278.

- Cui, L. *et al.* (2015) ‘Vibration response mechanism of faulty outer race rolling element bearings for quantitative analysis’, *Journal of Sound and Vibration*, pp. 1–10.
- Eftekharijad, B. *et al.* (2010) ‘The application of spectral kurtosis on Acoustic Emission and vibrations from a defective bearing’, *Mechanical Systems and Signal Processing*, 25, pp. 266–284.
- El-Thalji, I. and Jantunen, E. (2015) ‘Fault analysis of the wear fault development in rolling bearings’.
- French, M. L. and Hannon, W. M. (2015) ‘Angular contact ball bearing experimental spall propagation observations’, *Journal of Engineering Tribology*, 229(8), pp. 902–916.
- Gazizulin, D. *et al.* (2020) ‘A new efficient rolling element – Spall edge interaction model’, *International Journal of Fatigue*, 131, p. 105330.
- Gazizulin, D., Klein, R. and Bortman, J. (2017) ‘Towards efficient spall generation simulation in rolling element bearing’, *Fatigue and Fracture of Engineering Materials and Structures*, 40(9), pp. 1389–1405.
- Hill, K. O. and Meltz, G. (1997) ‘Fiber Bragg grating technology fundamentals and overview’, *Journal of Lightwave Technology*, 15(8), pp. 1263–1276.
- Jalalahmadi, B. *et al.* (2009) ‘A Review of Rolling Contact Fatigue’, *Article in Journal of Tribology*.
- Khmelnitsky, M. *et al.* (2015) ‘Improved bearing sensing for prognostics: From vibrations to optical fibres’, in *Insight: Non-Destructive Testing and Condition Monitoring*. British Institute of Non-Destructive Testing, pp. 437–441.
- Klein, R. *et al.* (2011) ‘Emphasising bearing tones for prognostics’, *International Journal of Condition Monitoring*, 1(2), pp. 73–78.
- Kogan, G. *et al.* (2015) ‘Toward a 3D dynamic model of a faulty duplex ball bearing’, *Mechanical Systems and Signal Processing*, 54, pp. 243–258.
- Kogan, G., Bortman, J. and Klein, R. (2017) ‘A new model for spall-rolling-element interaction’, *Nonlinear Dynamics*, 87(1), pp. 219–236.
- Qing, C. *et al.* (1991) ‘Measurement of the critical size

of inclusions initiating contact fatigue cracks and its application in bearing steel’, *Wear*, 147, pp. 285–294.

- Raje, N., Slack, T. and Sadeghi, F. (2009) ‘A discrete damage mechanics model for high cycle fatigue in polycrystalline materials subject to rolling contact’, *International Journal of Fatigue*, 31(2), pp. 346–360.
- Randall, R. B. (2011) ‘Vibration-based Condition Monitoring: Industrial, Automotive and Aerospace Applications.’, p. 309.
- Randall, R. B. and Rôme Antoni, J. (2010) ‘Rolling element bearing diagnostics-A tutorial’.
- Rosado, L. *et al.* (2010) ‘Rolling contact fatigue life and spall propagation of AISI M50, M50NiL, and AISI 52100, part I: Experimental results’, *Tribology Transactions*, 53(1), pp. 29–41.
- Tarawneh, C. *et al.* (2019) ‘Prognostics Models for Railroad Tapered Roller Bearings with Spall Defects on Inner or Outer Rings’, *Tribology Transactions*, 62(5), pp. 897–906.
- W.S. Siew, W.A. Smith, Z. Peng, R. B. R. (2015) ‘Fault severity trending in rolling element bearings - University of New South Wales’, in *Acoustics*.
- Wang, W., Sawalhi, N. and Becker, A. (2016) ‘Size Estimation for Naturally Occurring Bearing Faults Using Synchronous Averaging of Vibration Signals’, *Journal of Vibration and Acoustics, Transactions of the ASME*, 138(5).
- Wei, P. *et al.* (2016) ‘Fault diagnosis of the rolling bearing with optical fiber Bragg grating vibration sensor’, in Han, S. and Tan, J. (eds) *Optical Measurement Technology and Instrumentation*. SPIE, p. 101552I.
- Zhang, H. *et al.* (2021) ‘Tracking the natural evolution of bearing spall size using cyclic natural frequency perturbations in vibration signals’, *Mechanical Systems and Signal Processing*, 151.
- Zhang, H. *et al.* (2022) ‘A benchmark of measurement approaches to track the natural evolution of spall severity in rolling element bearings’, *MSSP*, 166, p. 108466.
- Zhen, L. *et al.* (2008) ‘Bearing condition monitoring based on shock pulse method and improved redundant lifting scheme’, *Mathematics and Computers in Simulation*, 79, pp. 318–338.

Ravit Ohana received her B.S. and M.S. degree in Mechanical Engineering from the Ben-Gurion University of the Negev. Currently, she is a PhD student. Her study focuses on spall propagation mechanisms of rolling element bearing by using physical based models and metallurgical analysis. Her main areas of research interest are rolling contact fatigue, finite element modeling, Acceleration and strain signals diagnostics and prognostics systems, and endurance tests.

Dr. Renata Klein received her B.Sc. in Physics and Ph.D. in the field of Signal Processing from the Technion, Israel Institute of Technology. In the first 17 years of her professional career, she worked in ADA-Rafael, the Israeli Armament Development Authority, where she managed the Vibration Analysis department. In the decade that followed, she focused on development of vibration based health management systems for machinery. She invented and managed the development of vibration based diagnostics and prognostics systems that are used successfully in combat helicopters of the Israeli Air Force, in UAVs and in jet engines. Renata is a lecturer in the faculty of Aerospace Engineering of the Technion, and in the faculty of Mechanical Engineering in Ben Gurion University of the Negev. In the recent years, Renata is the CEO and owner of R.K. Diagnostics, providing R&D services and algorithms to companies who wish to integrate Machinery health management and prognostics capabilities in their products.

Prof. Jacob Bortman joined the academic faculty of Ben-Gurion University of the Negev in September 2010 as a full Professor. Prof. Bortman spent thirty years in the Israel Air Force (IAF), retiring with rank of Brigadier General. His areas of research in the Dept. of Mechanical Engineering include: Health usage monitoring systems (HUMS); Conditioned based maintenance (CBM); Usage and fatigue damage survey; Finite Element Method; and Composite materials.

A Comparative Study of Health Monitoring Sensors based on Prognostic Performance

Hyung Jun Park¹, Nam Ho Kim², and Joo-Ho Choi³

¹*Dept. of Smart Air Mobility, Korea Aerospace University, Goyang-si, Gyeonggi-do, 10540, Korea*
phi921029@kau.kr

²*Dept. of Mechanical and Aerospace Engineering, University of Florida, Gainesville, Florida, FL 32611, USA*
nkim@ufl.edu

³*School. of Aerospace & Mechanical Engineering, Korea Aerospace University, Goyang-si, Gyeonggi-do, 10540, Korea*
jhchoi@kau.ac.kr

ABSTRACT

In the safety critical systems such as industrial plants or aircraft, failure occurs inevitably during the operation, and it is important to prevent this while maintaining high availability. Therefore, a lot of efforts are being directed toward developing advanced prognostics algorithms and sensing techniques as an enabler for predictive maintenance. The key for reliable and accurate prediction not only relies on the prognostics algorithms but also based on the collection of sensor data. However, there is not much in-depth studies toward evaluating the varying sensing techniques based on the prediction performance and inspection scheduling. It would be more reasonable for practitioner to select different cost of sensors based on the sensors' contribution on reducing the cost on unnecessary inspection or measurement while maintaining its prognosis performance. Thus, the authors try to thoroughly evaluate the cost-effectiveness of the different sensor with respect to sensor resistance to noise. The simulation is conducted to analyze the prediction performance with varying measurement interval and different level of noise during degradation. Then real run-to-fail (RTF) dataset acquired from two different sensors are analyzed to design optimal measurement system for predictive maintenance.

1. INTRODUCTION

To prevent catastrophic event due to safety system failure, the Prognostics and Health Management (PHM) techniques have been thoroughly studied to monitor the system health status and enable preventive maintenance. One of the key

enablers for reliable health monitoring is capturing and storing different kinds of data from various sensors that contain health condition information of the monitored equipment (Lei et al., 2018). From the measured data, practitioner can determine the current health condition using signal processing techniques, feature engineering, machine learning methods, etc. to further predict its remaining useful life (RUL) until failure based on various prognostics algorithms. The recent developments in sensing technology have provided numerous types of sensors to measure parameters such as acceleration, temperature, acoustic signals and etc. (Kalsoom et al., 2020). Acquiring high-quality information from various sensor types are more helpful for effective condition monitoring and prognostics. However, implementing great amount of sensors used in health monitoring research is impractical as it require a large amount of data storage and sensor implementation costs (Cheng, Azarian and Pecht, 2010). Therefore, a robust evaluation and guideline of each sensor capacity for health monitoring and prognostics need to be established.

The general method for sensor evaluation and selection is to select the degradation-relevant sensors which are adequate for prognostics. Liu et al (Liu et al., 2015) proposed entropy-based strategy to quantitatively select sensors that reflect the monotonic trend during degradation to perform engine health prognosis. Zhang et al (Zhang et al., 2020) and Coble et al (Coble and Hines, 2011) developed an additional selection metric considering the trend consistency of sensor data among different systems and validated with engine simulation datasets. The existing literatures for sensor selections are mainly focused on evaluating the sensors data to the degradation trend using metrics of monotonicity, correlation and robustness (Li et al., 2015; Zhang, Zhang and Xu, 2016; Liu et al., 2017; She and Jia, 2021). However, there is lack of sensor evaluation

Hyung Jun Park et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

toward actual prognosis performance with the applicability of prognostics algorithm. Moreover, most of the above studies aim to optimize the sensor network in case where a large amount sensor are installed and reduce the number of sensors for cost-effective prognosis. In this paper, authors focus more on analyzing and comparing typical sensors having different signal quality (level of noise interference) on prognosis performance and relate sensor costs on the effect of reducing amount of data during degradation.

Few studies have considered applicability of prognostics algorithms with the selection of different sensors. Camci et al (Camci et al., 2016) focused on developing guideline for practitioners on various types of sensors used to monitor the status of railway turnout system. The RUL performances of sensors have been evaluated and additional cost factors of sensors are considered to make an economic justification of the optimal sensor selection. Nevertheless, the metrics to evaluate prognosis performance is based on the true degradation information (Saxena et al., 2010; Yang et al., 2016) such as true End of Life (EOL), true degradation curve, the availability of historical run-to-fail data, etc. However, in practice, the degradation trend and noise differ by each sensor, and it is challenging to assess true information.

Motivated by the above issues, this paper evaluates the prognosis performance of sensors having different level of noise interference during degradation. In more detail, the most common and cost-effective contact type sensor, accelerometer (Lee et al., 2014) is considered as one. An acoustic non-contact sensor, microphone is considered as another since it is recently drawing attention as an alternative due to its advantage of low interference with external noises within the system (Park et al., 2021). For the performance evaluation, the authors utilize the metric that does not require true degradation information and validate its correlation with true information-based metric through numerical study. Finally, the prognosis performance under different amount of data (different data acquisition interval) during degradation is addressed to validate the cost of higher quality sensor.

2. METHODOLOGY

An overall framework of the study is described in Fig. 1. First, the design parameters related to sensor such as level of noise and data interval (data amount) are changed to generate various case of degradation datasets. Then, true information-based RUL performance and time window metric without true information are calculated from degradation datasets. Finally, correlation between two metrics are evaluated to validate the use of time window strategy and the capability to reduce data amount while maintaining prediction performance is verified. Finally, the time window metric is used in the bearing run-to-fail (RTF) datasets to evaluate the performance of two different quality

sensors and cost-effectiveness of high-quality sensor is analyzed.

2.1. Degradation simulation

In the simulation, the degradation function is assumed to follow an exponential function since various components such as battery and bearing degradation are widely known to degrade exponentially (An, Choi and Kim, 2013a; Kim et al., 2020; Lim et al., 2020). Thus, using degradation function defined as Eq. (1), data until 100 cycles are generated by changing two design parameters. Four different level of noise is added to the degradation dataset by uniform distribution and four data interval is used to vary data amount during degradation which is summarized in table 1.

$$y_j = e^{(bt_j)} + \sigma, b = 0.02 \quad (1)$$

Table 1. Sensor design parameters

Lv. of noise	$\sigma \sim U(-Lv.noise, Lv.noise)$ $Lv.noise = [0.2, 0.3, 0.4, 0.5]$
Data interval	$\Delta t = t_j - t_{j-1}, \Delta t = [1, 2, 4, 8]$

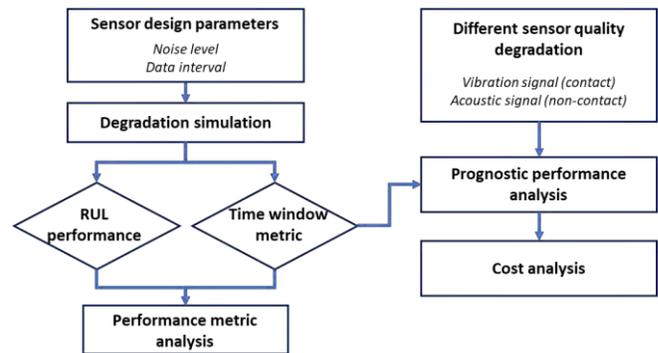
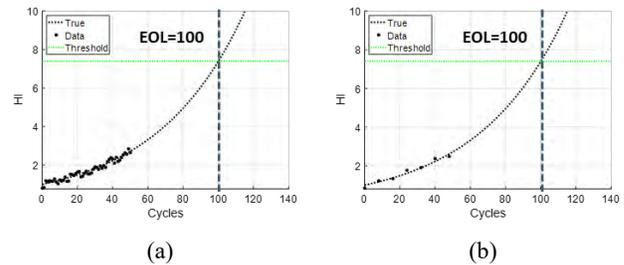


Figure 1. Overall framework of sensor evaluation



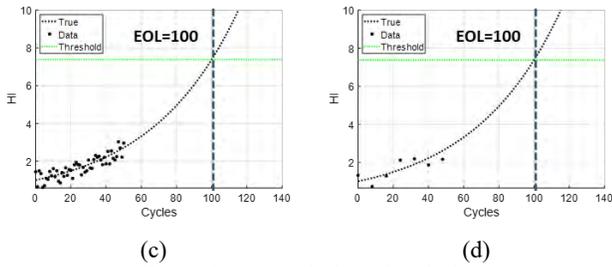


Figure 2. Degradation simulations

Based on the design parameters, total 16 cases are considered. The Fig. 2 (a) shows an example of dataset with small level of noise(0.2) and large amount of data($\Delta t = 1$). The black dots and dashed curved refer to measured data until 50 cycles and true degradation curve until end-of-life (EOL). A green line is the true threshold value at EOL which is used to calculate the predicted RUL. Fig. 2(b) is the case of small amount of data($\Delta t = 8$) and Fig. 2(c) and (d) are case under large level of noise(0.5). To consider the randomness of noise, 50 datasets are randomly generated under same design parameter.

2.2. Regularized Particle Filter (RPF)

Particle filter (PF) algorithm, also known as the Sequential Monte Carlo method is widely used prognostics approach in many engineering problems such as Lithium-ion batteries, induction motor and PEM fuel cells. PF recursively estimates and updates the probability distribution function (pdf) of the unknown model parameters or states of interest based on the following Bayes' theorem:

$$p(\theta|z) \propto L(z|\theta)p(\theta) \quad (2)$$

where θ is a vector of unknown parameters, z is a vector of measurements, $L(z|\theta)$ is the likelihood, $p(\theta)$ is the prior pdf of θ and $p(\theta|z)$ is the posterior pdf of θ conditional on z . Standard PF consists of state transition function f to predict the evolution of the state and measurement function h as follows:

$$x_k = f(x_{k-1}, b_k, v_k) = \exp(b_k dt)x_{k-1} \quad (3)$$

$$z_k = h(x_k, n_k) \quad (4)$$

where k is the time step index, x_k is the state, b_k is the vector of model parameter, z_k is the measurement data, and v_k and n_k are the process and measurement noises, respectively.

In this study, the exponential function is used for transition function and process noise is ignored since it can be handled through the uncertainty in the model parameters. For measurement, it is assumed that z_k is the same as degradation data including measurement noise having Gaussian noise, $n_k \sim N(0, \sigma_k)$, where σ_k is the unknown parameter estimated over time. Thus, the total unknown parameters to be estimated are $\theta = [x, b, \sigma]^T$.

The process of the PF is composed of three steps at each iteration. First, in prediction step, propagates the previous time step particles through state function to form particles at the current time, which is the prior pdf $p(\theta)$ at the current time. Then in the updating step, the likelihood of measurement data $L(z|\theta)$ that represents each particle's weight are calculated. As a new measurement is used, the weight of each particle is adjusted and assign a higher weight to the particles having a higher similarity with the measurement. Finally, in the resampling step, the particles are rearranged based on the obtained likelihood, which are duplicated or eliminated depending on the weight of the particles by using the inverse cumulative distribution function (CDF) method (Saha, Goebel and Christophersen, 2009; Dong et al., 2014). The resampled particles, which are the posterior distribution at the current time are then used as the initial distribution for the next step prediction. More information about PF is referred to (An, Choi and Kim, 2013b).

However, due to resampling process, PF-based prognosis suffers the problem of particle impoverishment since the samples are drawn from a discrete distribution rather than a continuous one. Consequently, after several iterations, the particles with small weights are discarded and the particles with high weights are duplicated too often which gives a poor representation of the posterior density. To resolve this issue, this study used regularized Particle filter (RPF) which is a modified version of PF in the resampling step. The kernels are generated at each particle points and summed to generate the kernel density estimate in RPF to have the advantage of approximating the weighted particles in continuous distributions (Musso, Oudjane and Legland, 2000).

2.3. Prognosis Performance Metric

After predicting the future degradation using the RPF algorithm, two different prognostic performance metrics are calculated: RUL performance metric and time window metric. To compare the performance of both metrics, two components are considered: the measure of the prediction accuracy and the measure of the uncertainty associated with the prediction. The schematic illustration of each metric calculation is addressed in Fig 3.

The RUL performance metric is calculated based on the result shown in the left figure of Fig. 3. The black dots and black dashed line represent the measured data until current cycle ($M = 50cycles$) and true degradation curve respectively. Measurement data until M cycles are used for estimating the distribution of model parameters. The red dashed line and light red colored space denote the predicted median and 90% confidence interval (C.I.) in the future. The green dotted line horizontally is the true threshold until failure which is the value corresponding to EOL (100 cycles). Based on the threshold, we can obtain the distribution of cycles when the predicted state reaches the

threshold. Then the distribution of RUL can be obtained by subtracting this pdf from current cycle which is 50 cycles. From the predicted RUL distribution, the accuracy measure can be calculated by Eq (5) which is an absolute error between median value of predicted RUL and true RUL. Besides accurate prediction, the level of uncertainty associated with the prediction is also an important factor to assess the prognostics performance from a conservative decision-making point of view. Therefore, the level of uncertainty is considered as normalized C.I. width which is

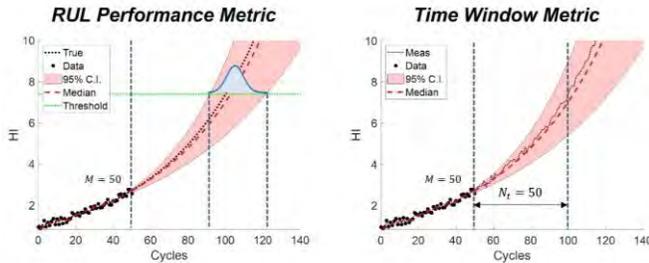


Figure 3. Illustration of two different prediction performance strategy

defined as RUL_{CI} and calculated by Eq. (6).

$$RUL_{error} = |RUL_{pred,median} - RUL_{true}| \quad (5)$$

$$RUL_{CI} = \frac{RUL_{est,5th} - RUL_{est,95th}}{RUL_{true}} \quad (6)$$

The time window metric is different from RUL metric as it directly uses the measurements in a certain time window without true information from degradation (Wang *et al.*, 2019). The strategy is shown in the right figure of Fig. 3 and the black dashed line is not a true degradation curve, but the measurement data. Since the true crack size is not available in practice, a straightforward way is to compare the predictions with data (Lei *et al.*, 2018; Wang *et al.*, 2019). Thus, the prediction accuracy and level of uncertainty are assessed over the time window which is the range between two vertical dashed line in the right figure. The length of time window (N_t) is set based on how much early prediction is required for maintenance scheduling. In the simulation study, the length of window is set 50 cycles same as the true RUL. The normalized mean square discrepancy (NMSD) is calculated to assess the prediction accuracy using below equation:

$$NMSD = \frac{1}{\max(y_{M+1:N_t}) - \min(y_{M+1:N_t})} MSD, \quad (7)$$

$$MSD = \frac{1}{N_t} \sum_{i=1}^{N_t} (y_{M+i} - \hat{x}^m_{M+i})^2$$

where M is the prediction start cycle, y is the measured data in the time window and \hat{x}^m is the predicted median of degradation state by the prognostic algorithm.

For the uncertainty measure, two indexes are considered together. The first index E_1 measures the relative width of the 90% C.I. with respect to the predicted median value for each cycle and averages over the time window, which is defined by Eq. (8). Thus, a smaller E_1 indicates a narrower C.I. over prediction and lower prediction uncertainty.

$$E_1 = \frac{1}{N_t} \sum_{i=1}^{N_t} \frac{(\hat{x}^u_{M+i} - \hat{x}^l_{M+i})}{\hat{x}^m_{M+i}} \quad (8)$$

\hat{x}^l & \hat{x}^u : Lower & upper bound of 90% C. I .

The second index E_2 measures whether the C.I. of prediction covers the true measurement and how wide the C.I. needs to be to cover the measurement at each prediction point. In detail, at each cycle in the time window, $M + i$, $i = 1, 2, \dots, N_t$, the minimal $\alpha\%$ C.I. that can cover the measure is calculated. The discrete values of α is increased from 90 to 99 with one increment and E_2 is calculated as $-0.01\alpha + 1$. A smaller α indicates a more reliable prediction, thus a larger value is assigned for E_2 . If the highest $\alpha = 99$ cannot cover the measurement, E_2 is defined zero.

$$E_2 = \begin{cases} \frac{1}{N_t} \sum_{i=1}^{N_t} -0.01\alpha_{M+i} + 1, \alpha = 90, 91, \dots, 99 \\ 0, \text{if required } \alpha \text{ exceeds } 99 \end{cases} \quad (9)$$

The above two indexes evaluate the prognostics uncertainty considering the C.I. of prediction. Based on each index characteristic, the smaller the E_1 and higher the E_2 represents better prediction. Thus, nonlinear combination of E_1 and E_2 , $EI = E_2/E_1$ is calculated as one index to assess uncertainty performance for time window metric.

2.4. Numerical Case Study

In this section, we attempt to evaluate the correlation between the RUL performance metric and the time window metric. Since the prediction performance of prognostic algorithm can differ by randomness of noise even under the same noise level, the correlation is analyzed using multiple randomly simulated datasets (50 datasets as mentioned before). If it has high correlation, the time window metric can be used to assess the true prediction performance using only measurement data.

The correlation is calculated between each component in metrics, the accuracy measure and uncertainty measure. The scatter plot and correlation coefficient value between two metrics under different level of noise are shown in Fig. 4. The upper figures present the scatter plot between RUL_{error} and NMSD and the lower figures between RUL_{CI} and EI .

Correlation between accuracy measures show very high correlation regardless of noise levels, smaller the RUL error, smaller the NMSD value. Though the uncertainty measure show less correlation than the accuracy measure, correlation coefficient value shows over 0.5 regardless of noise. Note that it has negative value since having higher EI value means more reliable prediction which corresponds to narrow RUL CI.

We verified that the time window metric has high correlation with true RUL performance and can be used as prediction performance indicator when true information is

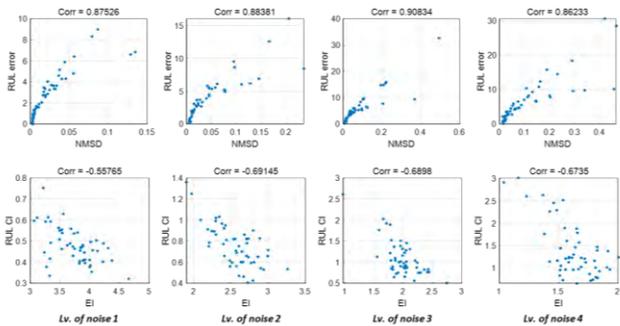


Figure 4. Correlation between two different metrics under varying noise

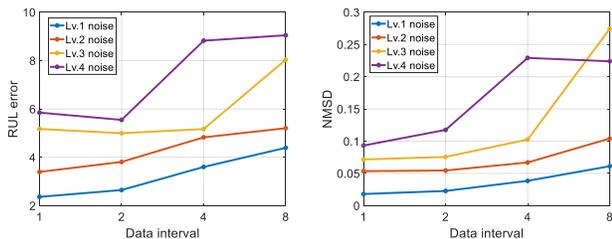


Figure 5. Prediction accuracy of dataset under varying level of noise with different data interval

unavailable. Moreover, the authors analyzed the prediction accuracy of dataset under varying level of noise with different data interval until current M cycles. Large data interval addresses small amount of data used for model parameter fitting. The overall result of RUL_{error} and NMSD are presented in Fig. 5 showing that measurement with small level of noise can have higher prediction accuracy than the measurement with larger noise while reducing the data amount. For example, the NMSD value at Lv. 1 noise & data interval of 4 cycles is smaller than the value at Lv. 3 noise & data interval of 1 cycle. Based on the verified hypothesis that metric without true information can assess the prediction performance and small level of noise data can maintain its performance higher than the high level of noise data, we evaluate the two different sensor data used on bearing health monitoring.

3. BEARING CASE STUDY

Bearing is one of the most critical components that leads to system failure and numerous researches have conducted to prevent its failure (Duong et al., 2018; Wang, 2018; Wu et al., 2019). Among the available sensors, the accelerometer has been the most common and cost-effective sensor for health monitoring. However, it has drawbacks of high interference with other signals due to its attachment within the system. Recently, acoustic non-contact sensor such as microphone are recently drawing attention as an alternative since it is less affected by the other signal interferences (Huang et al., 2019; Wang, Mao and Li, 2021).

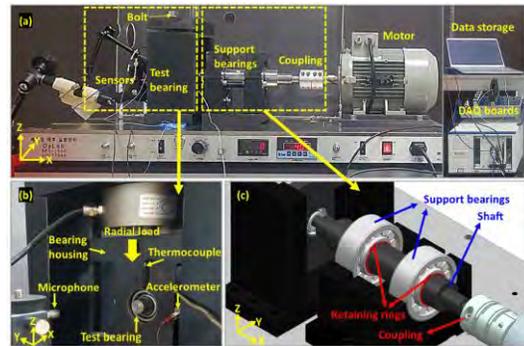


Figure 6. The bearing test rig: (a) Front view (b) Test bearing and sensors (c) Support bearings and couplings

To thoroughly evaluate the prognostics performance of two different quality sensors, the authors conducted multiple run-to-fail (RTF) experiment using a testbed mounted with commonly used accelerometer and high-quality microphone. The sensor performance regarding prognostics is evaluated using the time window metric from the previous section.

3.1. Experimental Setup

The Fig. 6 shows the bearing test rig in which the RTF tests are performed. The test rig consists of sensors, test bearings, support bearings, motor and DAQ boards. The sensors used in this study are an accelerometer (KS77C.100 by MMF) and a microphone (PCB Piezotronics 378C01) and a thermocouple. The cost of microphone (1931\$) is about 4 times higher than the accelerometer (452\$). The DAQ boards consist of NI Pxl-4464 and NI-9212, in which the former records acoustic and vibration signals at a sampling rate of 204.8 kHz and the latter records the temperature at a 100 Hz sampling rate. The first 1 second of every 10 seconds is stored as one cycle using LabVIEW software. The bearing is tested under the shaft rotation at 1700 rpm. Radial load generated by the mechanical fastening of bolts is applied to the test bearing located at the end of the shaft at 75~80% of the dynamic load rating of 7950N to develop natural growing defects. After a number of trials to ensure the faults fully developed over cycles while maintaining

safety, the test is terminated when the acoustic pressure, acceleration and temperature exceed the thresholds of 9 Pascal, 18 m/s^2 and 80°C, simultaneously. In result, three RTF test datasets are acquired by two different sensor signals having different level of noise and pattern during degradation.

3.2. Sensor Performance Analysis

The general process of fault prognosis requires to extract a proper health indicator (HI) for the prognostics. In this study, a traditional time-based statistical feature, the root mean square (RMS) is calculated as HI since it is widely used for bearing degradation monitoring. Though, there are various

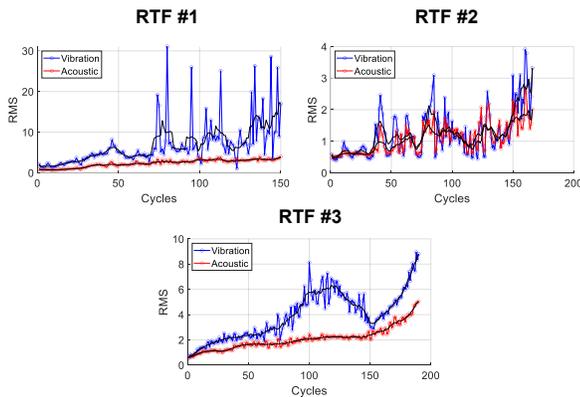


Figure 7. RMS values of three RTF datasets

studies that focused on developing robust HI for more accurate and earlier prediction, we use RMS since this article focuses more on evaluating the prognostics performance of different quality sensor data. The HIs for three different RTF datasets are presented in Fig. 7 where the blue and red dotted line is the HI from vibration signal and acoustic signal, respectively. It is noticeable that the vibration shows much fluctuation and noise interference compared to the acoustic signal. For the RTF #3, the HI from vibration shows not only large noise but also high fluctuation on the degradation trend itself.

The prognosis performance for each sensor is conducted using the time window metric. The window size (N_t) is fixed to 50 cycles in this study and prediction starting cycle (M) is 50 cycles which is used for PF model parameter fitting. Then the performance components representing the accuracy (NMSD) and uncertainty (EI) are calculated for every sequential cycle until failure. For example, the left and middle figures of Fig. 8 show the prediction result of RTF #1 dataset at each 50 cycles and 95 cycles, where upper figure is vibration and lower figure is acoustic. The black and red dots denote the measurement until current cycles and future measurement data in the time window. The red dashed line and light red colored space are the predicted median and 90% confidence interval (C.I.) in the

future. The NMSD and EI values over sequential cycle (from 50 cycles to 95 cycles) are shown at the right of Fig. 8 and averaged to evaluate the overall prognosis performance on each sensor data.

The prediction performance comparison of sensors for all RTF datasets are presented in Fig. 9. The upper histogram figures show the NMSD value, and the lower figures show the EI values related to uncertainty. The blue and red histogram are the prognosis metric results based on vibration and acoustic sensor, respectively. In addition, the results using data within interval of 4 cycles instead of 1 cycle are also compared together to evaluate the prognosis performance of acoustic sensor using less data amount. In the aspect of prediction accuracy, NMSD value of acoustic

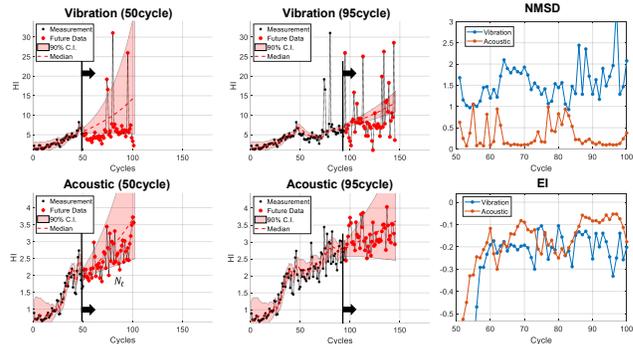


Figure 8. Prediction trajectories for each sensor and prognosis performance over degradation

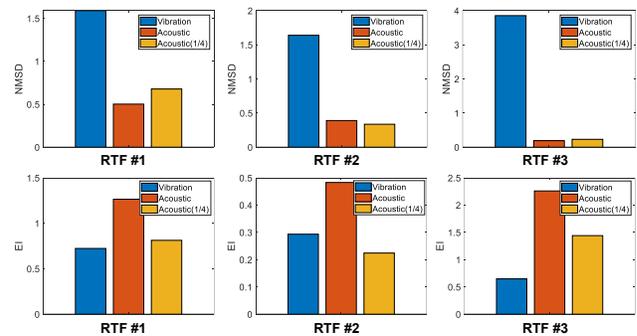


Figure 9. Prognosis performance comparison of vibration, acoustic and acoustic signal with less data (data interval of 4cycles)

is consistently lower than the vibration even if it uses 1/4 amount of data for model parameter fitting. The uncertainty measure EI shows that vibration sensor results lower reliability than the acoustic sensor over all RTF datasets. Though the acoustic EI values reduce when less data are used for the model parameter estimation, it is still similar or higher than the vibration. Thus, similar to the simulation study, the acoustic sensor, microphone having small level of noise during degradation perform much better prediction regardless of 1/4 times less data than the accelerometer.

Based on the prognosis performance comparison of two different sensor data, the effectiveness of microphone on prognostics is validated. Though the cost of microphone is higher than the conventional accelerometer, the performance on prediction accuracy and variation shows that microphone can provide much robust prediction even reducing the data storage costs. Considering the cost occurrence due to false prediction and data storage for long-term monitoring, the microphone will be cost-effective than implementing accelerometer.

4. CONCLUSION

A robust sensor performance comparison is proposed based on prognosis metric based only on direct measurement and without true degradation information. The validation of its metric is performance by randomly generating 50 datasets under different level of noise and calculating its correlation with true RUL performance. Then, addressed the advantage of using high quality sensor with less noise inference during degradation.

The bearing RTF experiments are conducted to demonstrate the two different quality sensors on prognosis performance. The non-contact sensor type, microphone showed superior performance on prognosis than the accelerometer due to the advantage of less interference to noise. Moreover, though the microphone costs much higher than the accelerometer, it is shown that it can reduce 4 times data amount than the accelerometer while maintaining its prognosis performance higher. This study is an initial step for setting a guideline for the practitioners when establishing the data acquisition system for PHM. The other factors such as calibration and regular maintenance of the instruments will be considered in the future work to evaluate sensor performance more robustly.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT) (No. 2020R1A4A407990411) and by the BK21 FOUR program through the National Research Foundation of Korea (NRF) funded by the Korean government (5199990714521).

REFERENCES

An, D., Choi, J. H., & Kim, N. H. (2013a). Options for Prognostics Methods: A review of data-driven and physics-based prognostics. In 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference (p. 1940).

An, D., Choi, J. H. and Kim, N. H. (2013b) ‘Prognostics 101: A tutorial for particle filter-based prognostics algorithm using Matlab’, *Reliability Engineering and System Safety*, 115, pp. 161–169. doi: 10.1016/j.res.2013.02.019.

Camci, F. *et al.* (2016) ‘Comparison of sensors and methodologies for effective prognostics on railway turnout systems’, *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 230(1), pp. 24–42. doi: 10.1177/0954409714525145.

Cheng, S., Azarian, M. H. and Pecht, M. G. (2010) ‘Sensor systems for prognostics and health management’, *Sensors*, 10(6), pp. 5774–5797. doi: 10.3390/s100605774.

Coble, J. and Hines, J. W. (2011) ‘Applying the general path model to estimation of remaining useful life’, *International Journal of Prognostics and Health Management*, 2(1), pp. 1–13.

Dong, H. *et al.* (2014) ‘Lithium-ion battery state of health monitoring and remaining useful life prediction based on support vector regression-particle filter’, *Journal of Power Sources*, 271, pp. 114–123. doi: 10.1016/j.jpowsour.2014.07.176.

Duong, B. P. *et al.* (2018) ‘A Reliable Health Indicator for Fault Prognosis of Bearings’. doi: 10.3390/s18113740.

Huang, H. *et al.* (2019) ‘Fault Diagnosis Accuracy Improvement Using Wayside Rectangular Microphone Array for Health Monitoring of Railway-Vehicle Wheel Bearing’, *IEEE Access*, 7, pp. 87410–87424. doi: 10.1109/ACCESS.2019.2924832.

Kalsoom, T. *et al.* (2020) ‘Advances in sensor technologies in the era of smart factory and industry 4.0’, *Sensors (Switzerland)*, 20(23), pp. 1–22. doi: 10.3390/s20236783.

Huang, H. *et al.* (2019) ‘Fault Diagnosis Accuracy Improvement Using Wayside Rectangular Microphone Array for Health Monitoring of Railway-Vehicle Wheel Bearing’, *IEEE Access*, 7, pp. 87410–87424. doi: 10.1109/ACCESS.2019.2924832.

Kim, S. *et al.* (2020) ‘A Novel Prognostics Approach using Shifting Kernel Particle Filter of Li-ion Batteries under State Changes’, *IEEE Transactions on Industrial Electronics*, 0046(c), pp. 1–1. doi: 10.1109/tie.2020.2978688.

Lee, J. *et al.* (2014) ‘Prognostics and health management design for rotary machinery systems - Reviews, methodology and applications’, *Mechanical Systems and Signal Processing*, 42(1–2), pp. 314–334. doi: 10.1016/j.ymsp.2013.06.004

Lei, Y. *et al.* (2018) ‘Machinery health prognostics: A systematic review from data acquisition to RUL prediction’, *Mechanical Systems and Signal Processing*. doi: 10.1016/j.ymsp.2017.11.016.

Li, N. *et al.* (2015) ‘An Improved Exponential Model for Predicting Remaining Useful Life of Rolling Element Bearings’, *IEEE Transactions on Industrial Electronics*, 62(12), pp. 7762–7773. doi: 10.1109/TIE.2015.2455055.

Lim, C. *et al.* (2020) ‘Feature extraction for bearing prognostics using weighted correlation of fault

- frequencies over cycles’, *Structural Health Monitoring*, p. 1475921719900917.
- Liu, L. et al. (2015) ‘Entropy-based sensor selection for condition monitoring and prognostics of aircraft engine’, *Microelectronics Reliability*, 55(9–10), pp. 2092–2096. doi: 10.1016/j.microrel.2015.06.076.
- Liu, L. et al. (2017) ‘Quantitative selection of sensor data based on improved permutation entropy for system remaining useful life prediction’, *Microelectronics Reliability*, 75, pp. 264–270. doi: 10.1016/j.microrel.2017.03.008.
- Musso, C., Oudjane, N. and Legland, F. (2000) ‘Improving Regularized Particle Filter’, *Sequential Monte-Carlo Method and Practice*, (April), pp. 247–271.
- Park, J. et al. (2021) ‘Frequency energy shift method for bearing fault prognosis using microphone sensor’, *Mechanical Systems and Signal Processing*. doi: 10.1016/j.ymssp.2020.107068.
- Saha, B., Goebel, K. and Christophersen, J. (2009) ‘Comparison of prognostic algorithms for estimating remaining useful life of batteries’, *Transactions of the Institute of Measurement and Control*. doi: 10.1177/0142331208092030.
- Saxena, A. et al. (2010) ‘Metrics for Offline Evaluation of Prognostic Performance’, *International Journal of Prognostics and Health Management (IJPHM)*, 1(1), pp. 1–20.
- She, D. and Jia, M. (2021) ‘A BiGRU method for remaining useful life prediction of machinery’, *Measurement: Journal of the International Measurement Confederation*, 167(3), pp. 1314–1326. doi: 10.1016/j.measurement.2020.108277.
- Wang, D. (2018) ‘Prognostics and Health Management : A Review of Vibration Based Bearing and Gear Health Indicators’, *IEEE Access*, 6, pp. 665–676. doi: 10.1109/ACCESS.2017.2774261.
- Wang, X., Mao, D. and Li, X. (2021) ‘Bearing fault diagnosis based on vibro-acoustic data fusion and 1D-CNN network’, *Measurement: Journal of the International Measurement Confederation*, 173(June 2020), p. 108518. doi: 10.1016/j.measurement.2020.108518.
- Wang, Y. et al. (2019) ‘Noise-dependent ranking of prognostics algorithms based on discrepancy without true damage information’, *Reliability Engineering and System Safety*, 184(October 2017), pp. 86–100. doi: 10.1016/j.res.2017.09.021.
- Wu, J. et al. (2019) ‘Degradation Data-Driven Time-To-Failure Prognostics Approach for Rolling Element Bearings in Electrical Machines’, *IEEE Transactions on Industrial Electronics*, 66(1), pp. 529–539. doi: 10.1109/TIE.2018.2811366.
- Yang, F. et al. (2016) ‘Health index-based prognostics for remaining useful life predictions in electrical machines’, *IEEE Transactions on Industrial Electronics*, 63(4), pp. 2633–2644. doi: 10.1109/TIE.2016.2515054.
- Zhang, B. et al. (2020) ‘Aircraft engine prognostics based on informative sensor selection and adaptive degradation modeling with functional principal component analysis’, *Sensors (Switzerland)*, 20(3). doi: 10.3390/s20030920.
- Zhang, B., Zhang, L. and Xu, J. (2016) ‘Degradation Feature Selection for Remaining Useful Life Prediction of Rolling Element Bearings’, *Quality and Reliability Engineering International*, 32(2), pp. 547–554. doi: 10.1002/qre.1771.

Forecasting piston rod seal failure based on acoustic emission features in ARIMA model

Jørgen. F. Pedersen¹, Rune Schlanbusch², Vignesh. V. Shanbhag³

^{1,2,3} Norwegian Research Centre, Energy & Technology Department, Jon Lilletuns Vei 9 H, 3. etg, 4879, Grimstad, Norway

jorgen.fone.pedersen@gmail.com

rusc@norceresearch.no

vigs@norceresearch.no

ABSTRACT

Fluid leakage due to piston rod seal failure in hydraulic cylinders results in unscheduled maintenance, machine downtime and loss of productivity. Therefore, it is vital to understand the piston rod seal failure at initial stages. In literature, very few attempts have been made to implement forecasting techniques for piston rod seal failure in hydraulic cylinders using acoustic emission (AE) features. Therefore, in this study, we aim to forecast piston rod seal failure using AE features in the auto regressive integrated moving average (ARIMA) model. AE features like root mean square (RMS) and mean absolute percentage error (MAPE) were collected from run-to-failure (RTF) tests that were conducted on a hydraulic test rig. The hydraulic test rig replicates the piston rod movement and fluid leakage conditions similar to what is normally observed in hydraulic cylinders. To assess reliability of our study, two RTF tests were conducted at 15 mm/s and 25 mm/s rod speed each. The process of seal wear from unworn to worn state in the hydraulic test rig was accelerated by creating longitudinal scratches on the piston rod. An ARIMA model was developed based on the RMS features that were calculated from four RTF tests. The ARIMA model can forecast the RMS values ahead in time as long as the original series does not experience any large shifts in variance or deviates heavily from the normal increasing trend. The ARIMA model provided good accuracy in forecasting the seal failure in at least two of four RTF tests that were conducted. The ARIMA model that was fitted with 15 pre-samples was used to forecast 10 out of sequence samples, and it showed a maximum moving absolute percentage error (MAPE value) of 28.99 % and a minimum of 4.950 %. The forecasting technique based on ARIMA model and AE features proposed in this study lays a strong basis to be used in industries to schedule the seal change in hydraulic cylinders.

KEYWORDS:

Hydraulic cylinder, Piston rod seal, Root mean square, Variable speed condition, Auto-regressive integrated moving average, Acoustic emission.

1. INTRODUCTION

A hydraulic cylinder is a linear actuator which is widely used in material handling applications in oil and gas (O&G), maritime, mining and construction industries. Based on the material handling requirements: load handling and speed condition of hydraulic cylinders frequently change. In most applications, customized large hydraulic cylinders are used by the industries where all the internal components are also custom-made (See ref. (“Large Hydraulic Cylinder”). Any abrupt failure of a hydraulic cylinder component can cause machine downtime, affect productivity, and increase maintenance cost as most of the components in large hydraulic cylinders are custom made and require several weeks time of planning, manufacturing, and assembling the part back into the hydraulic cylinder. Seal wear in hydraulic cylinders can be because of particle contaminants present in fluid or seal ageing and can cause instability during operation (X. Zhao et al. 2015; Shanbhag et al. 2021b). Therefore, it is important to continuously monitor and forecast the health of crucial components such as the piston rod seals in the hydraulic cylinders.

In recent years, acoustic emission sensors have been widely used to monitor fluid leakage due to seal wear in hydraulic cylinders. Acoustic emission (AE) sensors are preferred by researchers because of their high frequency range (0.5-2.5 MHz) which make them suitable to use in noisy or harsh environments, and they be used to simultaneously monitor the health of multiple components in hydraulic cylinders. For example, (Chen et al. 2007), monitored the health of seals in water hydraulic cylinders using time domain (root mean square (RMS) and count) and frequency domain (power spectral density (PSD)) features. Fluid leakage (< 1.0 L/min) due to seal wear could be monitored using energy-based

Jørgen. F. Pedersen et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

features (e.g., RMS). A correlation could be observed between fluid leakage rate and RMS. In the PSD plot, the fluid leakage was dominant in the frequency range of 50-300 kHz with a peak at 120 kHz. In the other work of (Petersen et al. 2005), monitored the health of the piston in a water hydraulic system using AE and wavelet analysis. RMS, PSD, and RMS of wavelet co-efficient were used to detect cracks in the piston rod. Using time domain feature RMS, it was possible to identify crack conditions in the piston rod. Compared to PSD, RMS calculated from wavelet co-efficient showed better separability between no-cracks and cracks in the piston rod. (Shanbhag et al. 2020), monitored the health of piston rod seals (unworn, semi-worn and worn conditions) on a customized hydraulic test rig using AE time-domain and frequency-domain features at different pressure conditions. It was observed that, the mean-frequency feature showed a good repeatability with sensitivity in identifying different seal wear conditions in the hydraulic test rig. In another work, (Shanbhag et al. 2021a) monitored the health of multiple components (piston rod seals and bearing strips) in the hydraulic test rig using AE time-domain and frequency-domain features to the bandpass filtered AE signal. Here, the unworn and worn bearing strips were monitored when unworn, semi-worn and worn seals were used in the test rig. The median-frequency features showed good repeatability in identifying piston rod seal wear and bearing wear conditions at different pressure and fluid leakage conditions. Also, mean-frequency and median-frequency showed good sensitivity in identifying fluid leakage due to piston rod seal wear during RTF tests (17 hours). (Zhang et al. 2021) monitored no leakage and different severities of fluid leakage (small, medium, and severe) in a hydraulic cylinder using an AE sensor. To classify the severity of fluid leakage, an optimization deep belief network (DBN) combined with the Complete Ensemble Empirical Model Decomposition with Adaptive Noise (CEEMDAN) was used and classification accuracy up to 93 % was achieved. (Pedersen et al. 2021), performed run-to-failure tests at different pressure and speed conditions on a hydraulic test rig to understand the AE features that can be evaluated to determine fluid leakage initiation. RMS features were proposed as potential condition monitoring indicators to understand fluid leakage initiation. The scaling factors based on sensor location and speed were applied to the sampled RMS features to estimate the fluid leakage threshold. From the literature, in the work performed using AE to monitor seal wear, most of the work is focused on condition monitoring (diagnostics) and very limited attempts in forecasting the deterioration and seal failure (prognostics).

The auto-regressive integrated moving average (ARIMA) model is a time series forecasting technique that is widely used in different applications such as disaster management, business forecasting, and machine prognostics. The ARIMA model can be used to understand the change in signal features with spatial heterogeneity over time (Li et al. 2021). In

literature, the ARIMA technique has been applied using AE features to predict a) energy change in gas-liquid two-phase flow (N. Zhao et al. 2021), b) coal and gas outburst (Li et al. 2021). As the ARIMA technique has successfully been used with the AE features for forecasting the process change or failure of components, the ARIMA technique in this research is used with AE features for forecasting the seal degradation. In this paper, the AE data from our previous experimental study conducted by (Pedersen et al. 2021) is used for forecasting analysis.

2. METHODOLOGY

2.1. Hydraulic test rig and process parameters

In this study, experiments were conducted on a test rig installed in an upright position (Figure 1) and was designed to replicate the fluid leakage conditions of a hydraulic cylinder. The test rig consists of three major items: a) test arrangement (electromechanical cylinder with pressure chamber), b) hydraulic system providing hydraulic power, c) control box which controls and monitors the test rig. The control box is connected to a laptop using an Ethernet cable and the test rig is controlled using the Bosch Rexroth software “IndraworksDs- 14.24.6”. A hydraulic power unit (HPU) supplies pressure to the pressure chamber in the test rig, which can be controlled using a pressure valve. The pressure chamber is connected to an electromechanical cylinder. The electromechanical cylinder consists of servomotor, spindle, and piston rod. The electromechanical cylinder uses a spindle and nut to convert rotational motion to translational motion. The servomotor drives the spindle, and the driven nut is connected to the piston rod. The piston rod in the test rig reciprocates through the pressure chamber that is made pressure tight using a typical rod-sealing concept. During the experiments, the chamber is pressurized while circulating medium (fluid) through the chamber to absorb heat and any debris caused by the seal wear.

In the test rig, five types of seals were used: a) wiper seal, b) excluder seal, c) secondary rod seal, d) primary rod low friction seal, and e) rod bearing ring. In this study, only the secondary rod seal and primary rod low friction seal were replaced with new seals during every test as the wear of these seals results in fluid leakage. Replacement took place during every test as the wear of these seals used to results in fluid leakage. Seal failure was defined when fluid leakage was observed from the leakage port in test rig. Typically, the seal life used in hydraulic cylinders in industry is for several years. However, in this study, the seal wear was accelerated by inducing scratches on the piston rod using a hard metal tip scribing tool. The process parameters used for the experiments are listed in Table 1.

Fluid in test rig	Water glycol
Rod material	Chromium-molybdenum steel (+QT) with 20µm HCr coating

Primary and secondary seal material	Polytetrafluoroethylene (PTFE)
Pressure	15 bar
Piston rod speed	15 mm/s (Test 2 & 3) and 25 mm/s (Test 1 & 4)
Number of tests	4
Test run time	Until fluid leakage observed
Stroke length	75 mm (Test 1 & 4); 150 mm (Test 2 & 3)

Table 1. Experimental details.

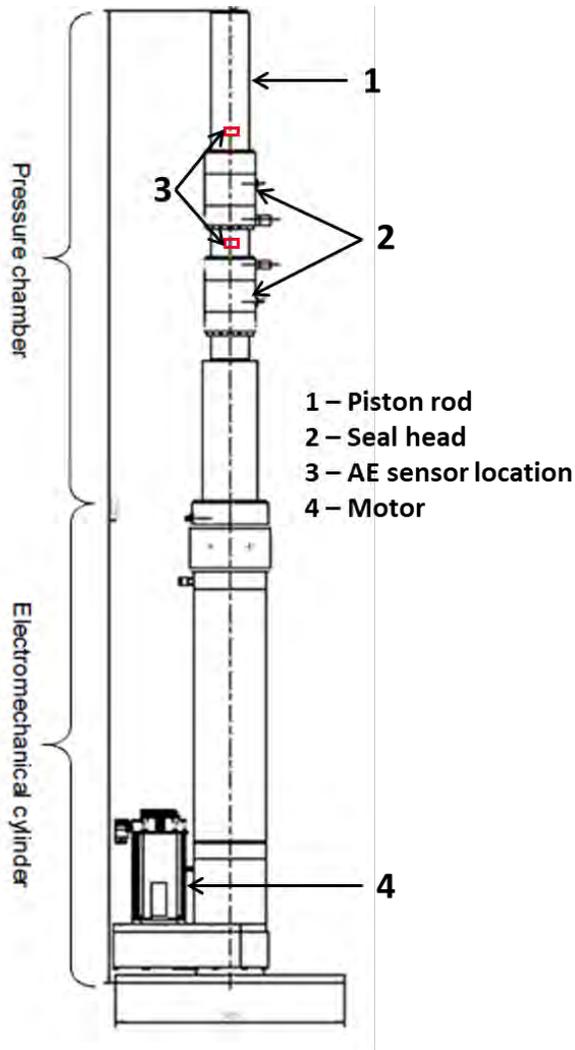


Figure 1. Schematic view of test rig.

2.2. Acoustic emission and signal processing

The AE sensor was mounted at two locations on the test rig: a) directly on the piston rod and, b) on the section of the cylinder below the seal head (see red squares indicating the positions in Figure 1). These two locations were selected as the measured AE signal energy was higher compared to other locations on the test rig. A mid-frequency range AE sensor with a frequency operating range of 50-400 kHz and resonant

frequency of 150 kHz was used in the study. The AE sensor was securely clamped on the test rig using an adhesive bond together with adhesive tape. The AE sensor was connected to a pre-amplifier and the pre-amplifier was further connected to an AE data acquisition system. The data acquisition system was connected to a laptop through a USB port. The AE data acquisition was performed using the Vallen AE suite software.

For all the experiments, the AE data acquisition was performed in continuous mode at a sampling rate of 1 MS/s and pre-amplifier gain of 40 dB. Due to the high sampling rate and the large size of the AE files, the AE data acquisition was limited to 90 seconds (five piston rod strokes) and data acquisition was performed at 15 minutes interval until the fluid leakage was observed. The AE signal was further analyzed using the MATLAB software. The AE signal of the extension and retraction strokes was observed to be similar (Figure 2). Therefore, only the AE signal from the extension stroke was used for forecasting analysis.

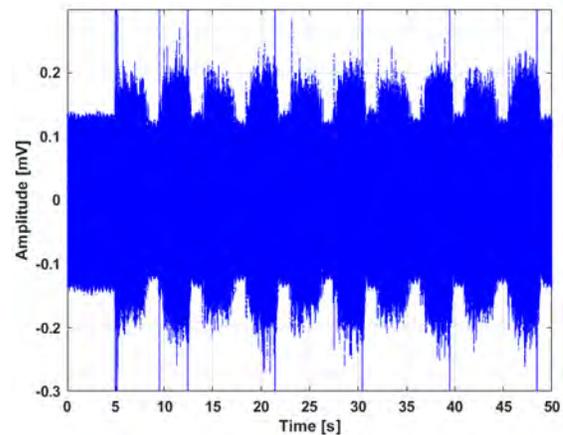


Figure 2. Raw AE signal recorded from test rig (Pedersen et al. 2021)

For every RTF test, a new piston rod seal was used in the seal head. Therefore, every test required the removal of the AE sensor from the test rig. To ensure, that the AE sensor clamping is consistent for every test, the Hsu-Nielsen pencil lead break test (See ref (“Acoustic Emission (AE): Hsu-Nielsen Source”)) was performed before the start of each test. The pencil lead break test was performed by breaking a 0.5 mm diameter pencil lead on the test rig surface near the mounted sensor. The amplitude of the AE burst response and magnitude of AE frequency response was calculated and compared during every test to ensure consistency of the AE sensor clamping on the test rig.

2.3. Auto-regressive moving average model

The ARIMA model is used in prediction of different types of time series data, e.g. financial or disaster prediction, as it can make the difference calculation in non-stationary time series data to form a stable series (Li et al. 2021). The ARIMA

model: a) auto-regressive (AR), b) integrated part (I), c) moving average part (MA). The ARIMA model is represented as ARIMA (p, d, q). Where, p is the order of the regressive model, d is the degree of difference, and q is the order of moving average model. The p, d, and q are used to make the model data as fit as possible. As per (Lee et al. 2011), the ARIMA model can be represented as a combination of past observations and past errors. The auto-regressive (AR) model uses past values in the time series to predict the future values in a time series. The AR model of order p, can be represented as:

$$x_n = \phi_1 x_{n-1} + \phi_2 x_{n-2} + \dots + \phi_p x_{n-p} + \omega_n \quad (1)$$

where in Eq. (1), x_n is the stationary time series, ω_n is Gaussian white noise series, and $\phi_1, \phi_2, \dots, \phi_p$ are the AR constants determined by an optimisation algorithm such as ordinary least squares (Shumway et al. 2017).

The moving average (MA) model uses its previous errors to make a prediction of future values. Here, the errors are the difference between the predicted value and the observed value. The MA model of order q, can be represented as:

$$x_n = \omega_n + \theta_1 \omega_{n-1} + \theta_2 \omega_{n-2} + \dots + \theta_q \omega_{n-q} \quad (2)$$

where in Eq. (2), ω_n is white noise, and $\theta_1, \theta_2, \dots, \theta_q$ are parameters (Shumway et al. 2017).

The integrated part (I) in the ARIMA model, means that the original timeseries are transformed from x_n to z_n via Eq. (3),

$$z_n = x_{n+1} - x_n \quad (3)$$

to make it stationary. The order of the integration parameter d is the order of difference performed on the time series.

2.3.1. Modelling of the condition monitoring data

In this study, the RTF test data was fitted using the ARIMA model. The Box-Jenkins model was used to select ARIMA (p, d, q) parameters and to validate the model fit. Each data set from the RTF test was used to fit in the ARIMA model to the most suitable condition monitoring data. To replicate a real-life condition, where the future data is unknown, only a portion of the initial samples were applied to fit the ARIMA model. The initial samples are labelled as pre-sample data. For creating the ARIMA model, fifteen samples from each RTF test were used as the pre-sample data. To test the accuracy of the developed ARIMA model, the next ten samples were used to forecast and to calculate the residual error. Based on the residual error, the root mean square error (RMSE) and the mean absolute percentage error (MAPE) was calculated as shown in Eq. (4). and Eq. (5).

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{x_i - \hat{x}_i}{x_i} \right| \times 100 \quad (4)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (x_i - \hat{x}_i)^2}{n}} \quad (5)$$

where, x_i is the true value, \hat{x} is the forecasted value, and n is the number of forecasted samples.

The auto-correlation function (ACF) and partial auto-correlation function (PACF) were used to graphically represent the relationship of a data point in a timeseries to data points from previous timesteps. These previous timesteps are called lags. Thus, a lag of one represents one timestep prior to the current timestep. Autocorrelation is then the calculated correlation between the current value and the values at the lags in a timeseries (Salvi 2019). Table 2 was used as a reference to determine preliminary values of the p and q parameters. The MATLAB in-built function was used to estimate the ARIMA (p, d, q) model from the pre-sample data. After estimating the model fitting parameters, the goodness of fit was validated by inferring the residuals from the fitted model. The selected ARIMA (p, d, q) model was then used to forecast the datapoints of the holdout data. The residuals were calculated from the known values of the holdout data and subtracting it from the forecasted values, and then the MAPE and RMSE were calculated. To compare the error values, the pre-sample data and holdout data were standardized by normalizing the values in the range of zero to one. To increase accuracy of the forecasted timeseries, a Monte Carlo simulation was applied to the forecasting timeseries. The Monte Carlo simulation used one thousand forecasting iterations with the pre-sample data as the input data. The mean of the forecasted predictor values was then used as the forecasted values.

	AR(p)	MA(q)	ARMA(p,q)
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

Table 2. Behavior of ACF and PACF for ARMA models (Shumway et al. 2017).

3. RESULTS AND DISCUSSION

3.1. Pencil lead break test

Figure 3 a)-b) represent the AE time domain signal of the background noise and from the pencil lead break test respectively. The AE signal of background noise was recorded while the HPU was circulating hydraulic fluid in the pressure chamber. By comparing Figure 3 a)-b), the maximum amplitude of the AE signal from the pencil lead break test is at least hundred times higher compared to the HPU background noise. Figure 3 c)-d), represent the AE frequency response calculated using Welch’s method. The frequency responses show that, the AE frequency peaks are dominant in the frequency range of 65-190 kHz. The maximum magnitude of the frequency response of the background noise is about one thousand times smaller than for the pencil break test. As the effect of ground noise on the AE signal is minimal, bandpass filtering techniques were not applied for the AE signal recorded during the RTF tests.

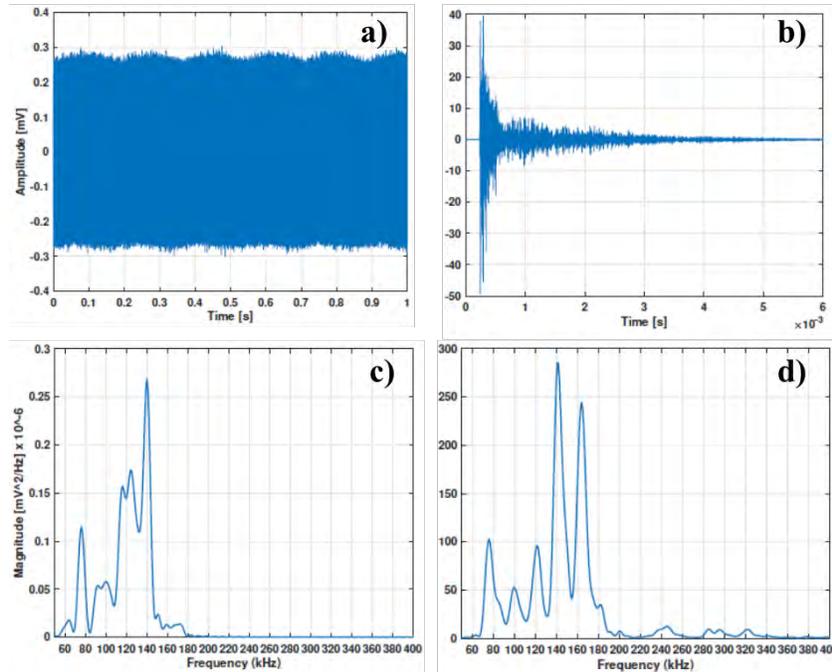


Figure 3. AE signal from a) Background noise, b) Pencil lead break test; Frequency response calculated from AE signal c) Background noise, d) Pencil lead break test.

3.2. ARIMA model using the RMS feature

From the RTF tests conducted a comparison of the time and frequency domain features were conducted, and it was observed that the RMS feature was the most suited for use as condition monitoring indicators to identify wear of piston rod seals (Pedersen et al. 2021). Therefore, in this study, the RMS feature was used to develop the ARIMA model. Figure 4-a) shows a plot of the RMS response for all four RTF tests. The signal was subtracted by the first sample to remove the bias and for easier comparison of the results. The increase in trend is similar for RTF tests 2 and 3 (tests conducted at 15 mm/s speed). For RTF 1 and 4, the trend shows a large difference (tests conducted at 25 mm/s speed). The drop in RMS feature in RTF 4, is mainly because the test was stopped at evening and restarted next day (in most industries hydraulic cylinders are used intermittently, not continuously). This has been done to observe the changes in signal response when the test rig was stopped. Tests 1 and 2 were run continuously, tests 3 and 4 were stopped in the night. In test 3, the next day system was switched on and kept running to allow system to be stabilized. Whereas in test 4 the next day, the system was switched on and data was recorded immediately to see the difference in behaviour of AE features with that of AE features from test 3. Furthermore, the transient response for the first three hours in RTF test 1 does not conform well to forecasting by the ARIMA model due to its initial decreasing trend. This is mainly because test rig pressure, and temperature require some time to stabilize. Therefore, for remaining tests, test rig was started only when test rig pressure and temperature were stabilized. To be able to do a better prediction on the RTF test

1 dataset, the transient response was removed. Figure 4-b) shows the responses for all RTF tests with the transient decreasing trend of RTF test 1 removed. As seen from Figure 4, the RMS feature trend is not stationary due to the increasing trend. To meet the stationary criteria of the ARIMA model, the RMS feature was differentiated. For RTF tests 1 and 3, a first order differentiation was applied, and for RTFs test 2 and 4, a second order differentiation was applied. Therefore, the differencing term ‘d’ in the ARIMA (p, d, q) was thus set as one for RTF tests 1 and 3, and two for RTF tests 2 and 4.

To identify the preliminary values of the AR (p) order, p, and MA (q) order, q, the ACF and PACF were plotted using the RMS features that were differentiated. Figure 5 shows the ACF and PACF plots for all differentiated data of the RTF tests. To find the initial parameters of the p, d, and q parameters for the ARIMA model, the guide in Table 2 was used to interpret the ACF and PACF plots. In Table 2, by “tailing off” it indicates the gradually decreasing correlation values, while the “cutting off” indicates the sudden large drop in correlation value. It can be seen in the PACF plot for RTF test 1 in Figure 5-e) that the PACF cuts off after the second lag. The ACF plot in Figure 5-a) does not show any lag above the threshold line, but it can be said to cut off after the first lag, even though the first lag is not very significant. An ARIMA (2,1,1) was thus suggested for the RMS signal from RTF test 1. For RTF test 2, both the ACF and the PACF plots in Figure 5-b) and Figure 5-f) show only one significant lag. However, the ACF plot can be seen to tail off while the PACF

plot cuts off at lag one. An ARIMA (1,2,0) model was thus suggested for the RMS signal from RTF test 2. The ACF plot for RTF test 3, in Figure 5, show very low correlation throughout the series, and only the second lag appears to show any correlation before it cuts off. The same can be seen for the PACF plot in Figure 5-g). Thus, to best model the RMS series for RTF test 3, an ARIMA (2,1,2) was suggested. Finally, for RTF test 4, the ACF plot in Figure 5-d) shows that it cuts off at the first lag. Similarly, the PACF plot in Figure 5-h) shows the same, but here the second lag can be seen to be more significant. Even though the second lag does not reach above the threshold line, it should still be utilized

in the model. An ARIMA (2,2,1) was thus suggested for the RMS series for RTF test 4.

The quantile-quantile (QQ) plots for the residuals of the fitted model on the pre-sample data is shown in Figure 6. It can be seen that all fitted models are reasonably normally distributed, except for the possible outliers as seen for the last quantile of RTF tests 1 and 4 in Figure 6-a) and Figure 6-d). The ACF and PACF plots of the residuals of the fitted models are represented in Figure 7. The models fitted to the RMS series for all RTF tests show a low correlation of the residuals both for the ACF and PACF. This indicates that the selected p , d , and q parameters provide good model fits to the data.

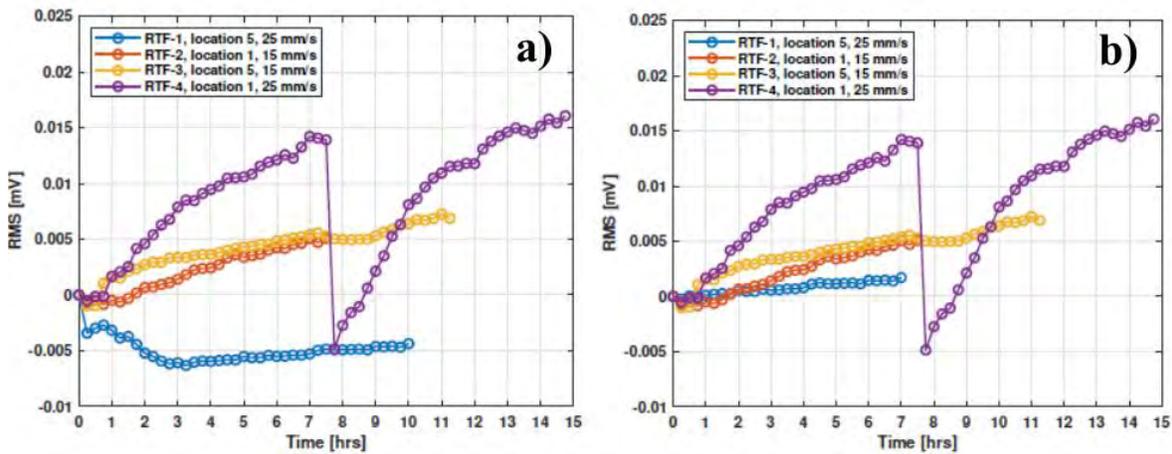


Figure 4. a) With transient from RTF test 1, b) Transient removed from RTF test 1.

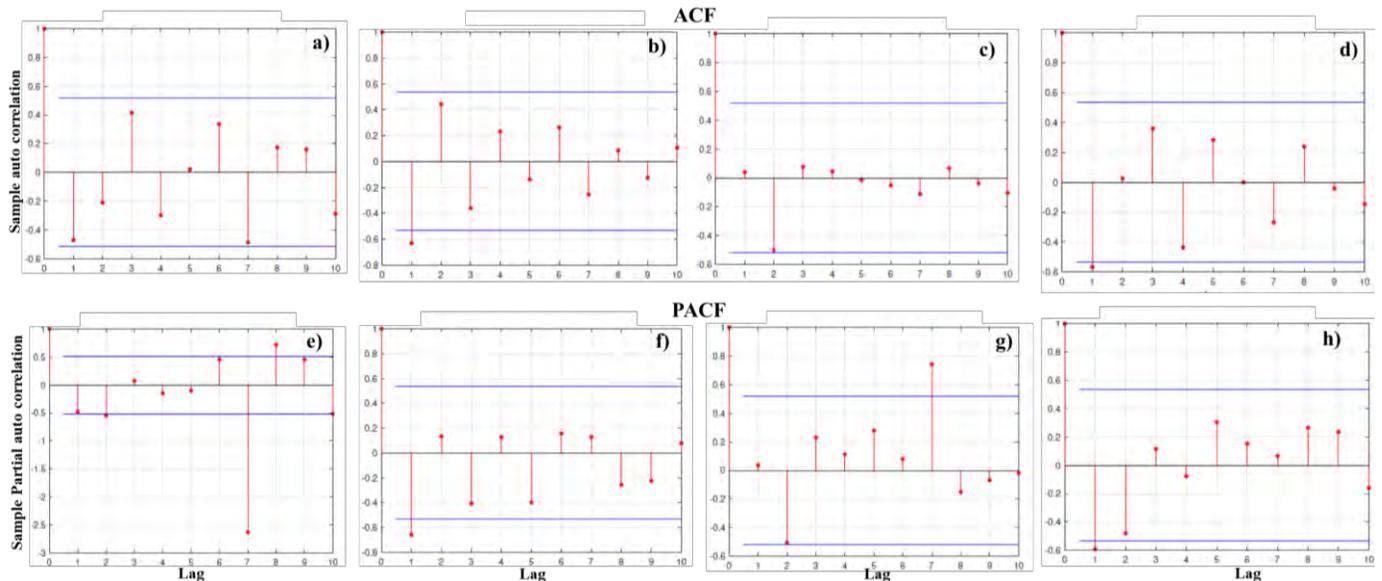


Figure 5. ACF and PACF for the differentiated RMS series of all RTF datasets, showing first 10 lags. a)-d) ACF, RMS signals from RTF 1-4, e)-f) PACF, RMS signals from RTF 1-4.

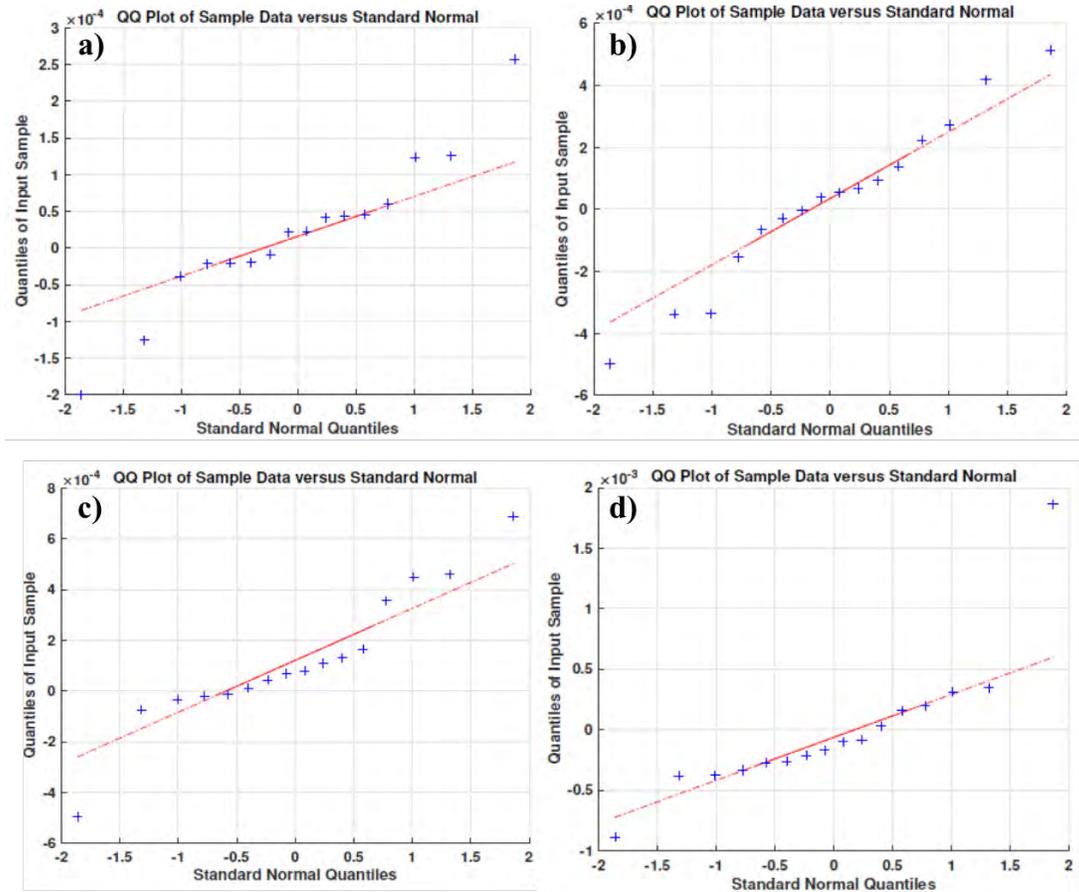


Figure 6. QQ plots for fitted ARIMA models to the RMS series of RTF tests 1 to 4: a) for ARIMA (2, 1, 1) model on RTF test 1, b) for ARIMA (1, 2, 0) model on RTF test 2, c) for ARIMA (2, 1, 2) model on RTF test 3, d) ARIMA (2, 2, 1) model on RTF test 4.

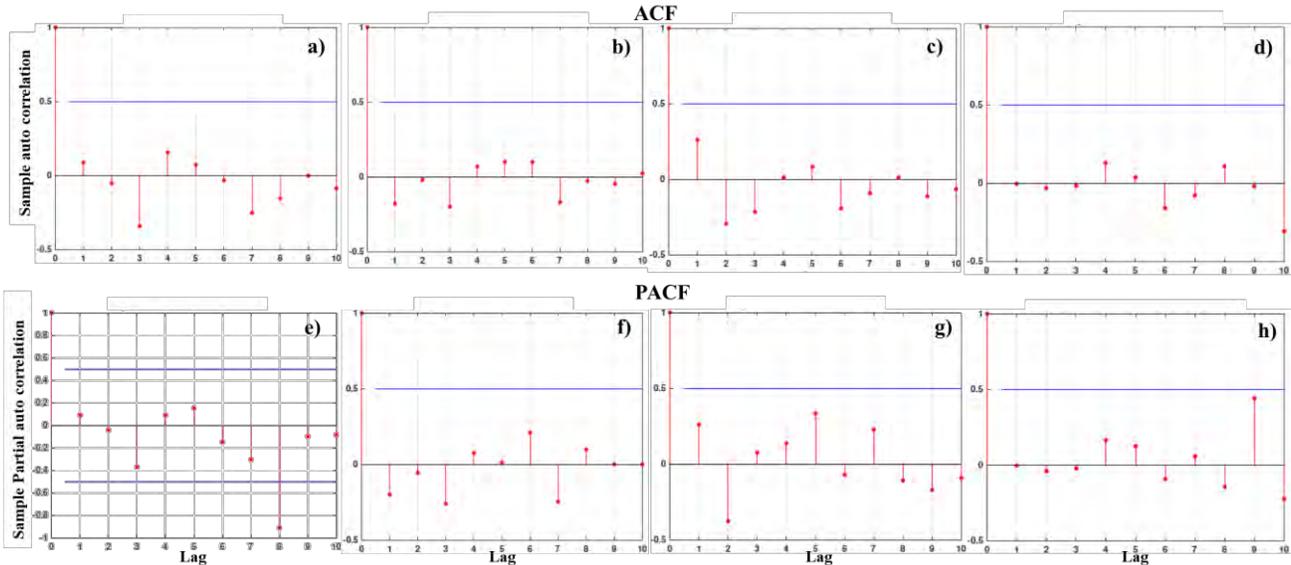


Figure 7. ACF and PACF for the residuals of the fitted ARIMA models on the pre-sample data. ACF for residuals of a) ARIMA (2, 1, 1) model on RTF test 1, b) ARIMA (1, 2, 0) model on RTF test 2, c) ARIMA (2, 1, 2) model on RTF test 3, d) ARIMA (2, 2, 1) model on RTF test 4. PACF for residuals of e) ARIMA (2, 1, 1) model on RTF test 1, f) ARIMA (1, 2, 0) model on RTF test 2, g) ARIMA (2, 1, 2) model on RTF test 3, h) ARIMA (2, 2, 1) model on RTF test 4.

3.3. Forecasting using the ARIMA model

Table 3 represents the best fitted ARIMA model parameters with the RMSE and MAPE values for the ten samples out of sequence forecasts. Figure 8 represents the forecasting of the RTF test data using the ARIMA model with the 95th percentile of the forecasts from the Monte-Carlo simulation. For all RTF tests, the forecasting plot can be seen to follow the increasing trend of the true data. Comparing the forecasting trend among the data from the RTF tests 1-4, for the RTF tests 1 and 2, see Figure 8 a)-b), the accuracy is less compared to RTF tests 3 and 4. The low accuracy of the forecast trend in RTF test 1 is mainly due to the large variance shift in the original dataset seen at around 4 hours, see Figure 4-a)). For the RTF test 2, the low accuracy for the ARIMA model is attributed to the low correlation of sequence that was

seen in the related ACF and PACF. For RTF tests 3 and 4, the ARIMA models displays good accuracy for the forecasted values, see Figure 8 c)-d) despite the low correlation of sequence also for these timeseries. The better accuracy of the model for RTF 3 and 4, can also be attributed to a favorable time of forecasting in the series.

RTF	AR (p)	I(d)	MA(q)	RMSE (mV)	MAPE (%)
1	2	1	1	0.187	20.26
2	1	2	0	0.326	28.99
3	2	1	2	0.053	4.95
4	2	2	1	0.104	8.88

Table 3. ARIMA (p, d, q) model parameters with the respective RMSE and MAPE error.

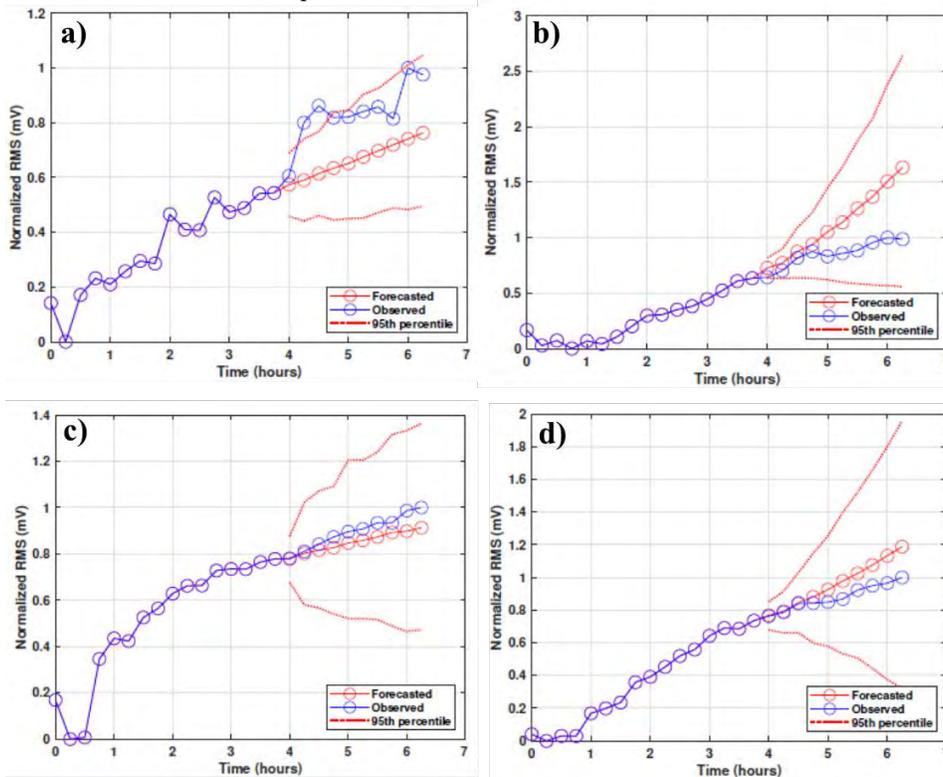


Figure 8. Forecasted data on 10 sample forecasts for RMS series of all RTF tests. a) For ARIMA (2,1,1) model on RTF test 1, b) For ARIMA (1,2,0) model on RTF test 2, c) For ARIMA (2,1,2) model on RTF test 3, d) For ARIMA (2,2,1) model on RTF test 4.

4. SUMMARY

In this study, the AE-RMS feature from four RTF tests was used to forecast the seal degradation process in a hydraulic test rig using an ARIMA model. The ARIMA model was able to forecast the RMS values ahead in time as long as the original RMS trend did not experience any large shifts in variance or deviates from the normal increasing trend, as is expected from this method. The ARIMA model showed that it can perform with good accuracy for forecasting in at least two of four RTF tests that were conducted. The ARIMA model that was fitted with fifteen pre-samples, was used to

forecast ten out of sequence samples, and it showed a maximum moving absolute percentage error (MAPE) a maximum of 28.99 % and a minimum of 4.950 %.

Based on the work conducted in this study, the authors conclude that further work is required with other modelling approaches like different variants of neural network for forecasting the seal failure, to improve the prediction when there are large shifts in variance that was seen in the RMS trend. Also, additional RTF tests need to be conducted with similar conditions to assess the repeatability of the forecasting technique.

Acknowledgement

The research presented in this paper has received funding from the Norwegian Research Council, SFI Offshore Mechatronics, project number 2378.

REFERENCES

- Chen, P., P.S.K. Chua, and G.H. Lim. 2007. "A Study of Hydraulic Seal Integrity." *Mechanical Systems and Signal Processing* 21 (2): 1115–26. <https://doi.org/10.1016/j.ymsp.2005.09.002>.
- "Large-Hydraulic-Cylinder-Brochure.Pdf." n.d. Accessed March 28, 2022. https://dc-corp.resource.bosch.com/media/general_use/products/industrial_hydraulics_1/cylinders_1/Large-Hydraulic-Cylinder-Brochure.pdf.
- Lee, Yi-Shian, and Lee-Ing Tong. 2011. "Forecasting Time Series Using a Methodology Based on Autoregressive Integrated Moving Average and Genetic Programming." *Knowledge-Based Systems* 24 (1): 66–72. <https://doi.org/10.1016/j.knosys.2010.07.006>.
- Li, Bing, Enyuan Wang, Zheng Shang, Xiaofei Liu, Zhonghui Li, Baolin Li, Hao Wang, Yue Niu, and Yue Song. 2021. "Optimize the Early Warning Time of Coal and Gas Outburst by Multi-Source Information Fusion Method during the Tunneling Process." *Process Safety and Environmental Protection* 149 (May): 839–49. <https://doi.org/10.1016/j.psep.2021.03.029>.
- "NDT Encyclopedia - Acoustic Emission (AE): Hsu-Nielsen Source." n.d. Accessed March 28, 2022. <https://www.ndt.net/article/az/ae/hsunielensource.htm>.
- Pedersen, Jørgen F., Rune Schlanbusch, Thomas J. J. Meyer, Leo W. Caspers, and Vignesh V. Shanbhag. 2021. "Acoustic Emission-Based Condition Monitoring and Remaining Useful Life Prediction of Hydraulic Cylinder Rod Seals." *Sensors* 21 (18): 6012. <https://doi.org/10.3390/s21186012>.
- Petersen, Dr, Re Link, P Chen, Psk Chua, and Gh Lim. 2005. "An Experimental Study of Monitoring Internal Leakage in Water Hydraulic Cylinders Using Acoustic Emission." *Journal of Testing and Evaluation* 33 (6): 12534. <https://doi.org/10.1520/JTE12534>.
- Salvi, Jayesh. 2019. "Significance of ACF and PACF Plots In Time Series Analysis." Medium. March 27, 2019. <https://towardsdatascience.com/significance-of-acf-and-pacf-plots-in-time-series-analysis-2fa11a5d10a8>.
- Shanbhag, Vignesh V., Thomas J. J. Meyer, Leo W. Caspers, and Rune Schlanbusch. 2020. "Condition Monitoring of Hydraulic Cylinder Seals Using Acoustic Emissions." *The International Journal of Advanced Manufacturing Technology* 109 (5–6): 1727–39. <https://doi.org/10.1007/s00170-020-05738-4>.
- Shanbhag, Vignesh V., Thomas J. J. Meyer, Leo W. Caspers, and Rune Schlanbusch. 2021a. "Defining Acoustic Emission-Based Condition Monitoring Indicators for Monitoring Piston Rod Seal and Bearing Wear in Hydraulic Cylinders." *The International Journal of Advanced Manufacturing Technology* 115 (9–10): 2729–46. <https://doi.org/10.1007/s00170-021-07340-8>.
- Shanbhag, Vignesh V., Thomas J. J. Meyer, Leo W. Caspers, and Rune Schlanbusch. 2021b. "Failure Monitoring and Predictive Maintenance of Hydraulic Cylinder—State-of-the-Art Review." *IEEE/ASME Transactions on Mechatronics* 26 (6): 3087–3103. <https://doi.org/10.1109/TMECH.2021.3053173>.
- Shumway, Robert H., and David S. Stoffer. 2017. *Time Series Analysis and Its Applications: With R Examples*. Springer Texts in Statistics. Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-319-52452-8>.
- Zhang, Peng, and Xinyuan Chen. 2021. "Internal Leakage Diagnosis of a Hydraulic Cylinder Based on Optimization DBN Using the CEEMDAN Technique." Edited by Li Qing. *Shock and Vibration* 2021 (March): 1–10. <https://doi.org/10.1155/2021/8856835>.
- Zhao, Ning, Chaofan Li, Huijun Jia, Fan Wang, Zhiyue Zhao, Lide Fang, and Xiaoting Li. 2021. "Acoustic Emission-Based Flow Noise Detection and Mechanism Analysis for Gas-Liquid Two-Phase Flow." *Measurement* 179 (July): 109480. <https://doi.org/10.1016/j.measurement.2021.109480>.
- Zhao, Xiuxu, Shuanshuan Zhang, Chuanli Zhou, Zhemin Hu, Rui Li, and Jihai Jiang. 2015. "Experimental Study of Hydraulic Cylinder Leakage and Fault Feature Extraction Based on Wavelet Packet Analysis." *Computers & Fluids* 106 (January): 33–40. <https://doi.org/10.1016/j.compfluid.2014.09.034>.

Improved time-frequency representation for non-stationary vibrations of slow rotating machinery

Cédric Peeters¹, Andreas Jakobsson², Jérôme Antoni³, and Jan Helsen¹

¹ *Department of Applied Mechanics, Vrije Universiteit Brussel, Brussels, Belgium*
cedric.peeters@vub.be
jan.helsen@vub.be

² *Center for Mathematical Sciences, Lund University, Sweden*
andreas.jakobsson@matstat.lu.se

³ *Univ Lyon, INSA Lyon, LVA, Villeurbanne, France*
jerome.antoni@insa-lyon.fr

ABSTRACT

The short-time Fourier transform (STFT) is a staple analysis tool for vibration signal processing due to it being a robust, non-parametric, and computationally efficient technique to analyze non-stationary signals. However, despite these beneficial properties, the STFT suffers from high variance, high sidelobes, and a low resolution. This paper investigates an alternative non-parametric method, namely the sliding-window iterative adaptive approach, to use for time-frequency representations of non-stationary vibrations. This method reduces the sidelobe levels and allows for high resolution estimates. The performance of the method is evaluated on both simulated and experimental vibration data of slow rotating machinery such as a multi-megawatt wind turbine gearbox. The results indicate significant benefits as compared to the STFT with regard to accuracy, readability, and versatility.

1. INTRODUCTION

Spectral analysis of vibration signals plays a crucial role in the majority of existing condition monitoring schemes. A commonly employed spectral analysis tool to investigate vibrations from machinery operating in non-stationary conditions is the visualisation of time-frequency representations (TFRs). These TFRs of vibration signals can be valuable for various reasons and have therefore been used for multiple different purposes. Example usages in vibration analysis include tracking the amplitudes of specific signal components over time (Sapena-Bano, Burriel-Valencia, Pineda-Sanchez, Puche-Panadero, & Riera-Guasp, 2016), assessing

the degree of non-stationarity (Martin & Mailhes, 2009), extracting rotating speed information (Peeters et al., 2019), separating asynchronous harmonics (Chen & Feng, 2021), and whitening the signal (Leclere, André, & Antoni, 2016). Research into improving existing TFR techniques has been ongoing and plentiful for the last few decades with many new techniques for TF decompositions having been developed, e.g. (Shensa et al., 1992; Barkat & Boashash, 1999; Greitans, 2005; Gardner & Magnasco, 2006; Wang, 2007; Wang & Orchard, 2009; Du, Li, Stoica, Ling, & Ram, 2009; Zhang & Castagna, 2011; Daubechies, Lu, & Wu, 2011; Oberlin, Meignen, & Perrier, 2014). While TFRs can be a precursor to other post-processing methods such as rotating speed estimation techniques, they can also serve as direct tools for fault detection. An example of such a use-case is the tracking of non-stationary transient signatures in a TFR over different frequency bands for bearing fault detection (Wang et al., 2019).

The analysis of multicomponent signals through TFRs is a much researched topic in the signal processing literature. One reason why there is so much literature about time-frequency (TF) analysis is the wide range of application domains, e.g. acoustics (Neal, Briggs, Raich, & Fern, 2011; Baydar & Ball, 2001), structural and machine health monitoring (Baydar & Ball, 2001; Feng, Liang, & Chu, 2013; He, 2013; Peng, Li, Hao, & Xin, 2020), physiological signals (Bozkurt, Germanakis, & Stylianou, 2018), astronomy (Liu, Zhang, & Shan, 2018), hydrology (Labat, 2005), seismology (Spanos, Giaralis, & Politis, 2007), climatology (Torrence & Compo, 1998; Salisbury & Wimbush, 2002; Kravchinsky, Langereis, Walker, Dlusskiy, & White, 2013), ecology (Cazelles et al., 2008), and geology (Reager, Thomas, & Famiglietti, 2014). Some common non-parametric TF techniques include

Cédric Peeters et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

the STFT (Flanagan & Golden, 1966), Wigner-Ville Distribution (WVD) (Wigner, 1932), Choi-Williams Distribution (CWD) (Choi & Williams, 1989), S-transform (Stockwell, Mansinha, & Lowe, 1996), and the chirplet transform (Mann & Haykin, 1991). These methods have all been used in the past for vibration analysis and monitoring as they form an appealing choice thanks to being hyperparameter-free. However, making no signal model assumptions can also be disadvantageous and most of these techniques are considered to have drawbacks depending on the intended application. This can be illustrated with the STFT, which uses a constant window size for both low and high frequencies, introducing a trade-off between time and frequency resolution. The efficacy of the STFT is thus hindered by the window choice as the Heisenberg uncertainty principle (Gabor, 1946) limits the achievable adaptability of the STFT. Similarly, the original definition of the WVD has an obvious downside thanks to the presence of large cross-terms between every pair of signal components and between positive and negative frequencies. The CWD suppresses the cross-terms of the WVD, but still suffers from aliasing for transients where a frequency component can be replicated at a distance π (Zheng & McFadden, 1999). While the S-transform is similar to the wavelet transform in that it offers a higher time resolution with lower frequency resolution at high frequencies and a higher frequency resolution with lower time resolution at low frequencies, it therefore also suffers from the same drawback that this resolution trade-off might not be desirable.

Despite the existence of this variety of TFR methods, probably the short-time Fourier transform (STFT) remains the most used conventional technique due to it being an easy-to-interpret non-parametric TFR method with low computational complexity and no model assumptions. The STFT is also reliable for the analysis of complex vibrations that contain an unknown number of non-stationary signal components with varying or low signal-to-noise ratios (SNR), a property which is not always shared by some other, typically parametric, developments that need a prior estimate of the number of signal components. For integration into an automated vibration processing methodology, the benefit of having a standalone method that does not require any data-dependent hyperparameter setting is quite a significant one. When large amounts of highly variable vibration data need to be processed, it is simply not feasible to optimize these hyperparameters for each dataset.

This paper investigates the potential of an adaptive spectral estimation alternative to the STFT method for vibration analysis that offers a reduction in leakage effects in exchange for a higher computational complexity. The sliding-window or short-time iterative adaptive approach (ST-IAA) is a high-resolution data-dependent filterbank-based approach that has been briefly investigated in the past for passive sensing and radar applications (Du et al., 2009) and also for human gait

analysis (Du et al., 2009). However, these applications involve vastly different signal complexities when compared to vibrations measured on complex machinery. This work therefore analyses the performance of the ST-IAA on such complex vibration signals with a focus on slow rotating machinery and it tries to lay the groundwork for more advanced applications in vibration analysis of the ST-IAA and similar techniques in the future. Section 2 introduces the theory behind the short-time iterative adaptive approach, whilst sections 3 and 4 illustrate the technique on realistic simulated vibration signals and experimental data, respectively. The results and next steps are discussed in section 5.

2. METHODOLOGY

The Iterative Adaptive Approach (IAA) is a spectral estimation technique that gained a lot of interest in the early years of the previous decade for the purpose of source localization, pulse compression, and missing data estimation (Yardibi, Li, Stoica, Xue, & Baggeroer, 2010; Karlsson, Rowe, Xu, Grentis, & Li, 2014). It is an iterative weighted least-squares method that is non-parametric and thus easy to use. It has been shown in the past that the IAA can reduce sidelobe levels and yield a higher resolution than the standard periodogram. Additionally, it also returns a dense (i.e., not sparse) estimate of the signal power spectrum which can be beneficial when dealing with complex vibrations, since enforcing sparsity typically involves parameter tuning. IAA assumes that the vibration data adheres to the following signal model:

$$\mathbf{y}_N = \mathbf{F}_{N,K} \boldsymbol{\alpha}_K + \mathbf{e}_N \quad (1)$$

with $\mathbf{y}_N \in \mathbb{R}^N$ being the vibration signal of length N , $\mathbf{F}_{N,K} \triangleq [\mathbf{f}_N(\omega_0), \mathbf{f}_N(\omega_1), \dots, \mathbf{f}_N(\omega_{K-1})]$ the Fourier matrix of size $(N \times K)$, $\boldsymbol{\alpha}_K \triangleq [\alpha(\omega_0), \alpha(\omega_1), \dots, \alpha(\omega_{K-1})]^T$ the complex-valued spectral amplitudes at the frequencies ω_k , and \mathbf{e}_N an additive noise. IAA tries to estimate $\boldsymbol{\alpha}_K$ from Eq. 1 by minimizing the following weighted least-squares cost function:

$$\|\mathbf{y}_N - \mathbf{f}_N(\omega_k) \alpha_k\|_{\mathbf{Q}_N^{-1}(\omega_k)}^2, k = 0, 1, \dots, K-1 \quad (2)$$

where $\|\mathbf{z}\|_{\mathbf{Q}_N^{-1}(\omega_k)}^2 \triangleq \mathbf{z}^H \mathbf{Q}_N^{-1}(\omega_k) \mathbf{z}$ and:

$$\mathbf{Q}_N(\omega_k) = \mathbf{R}_N - p_k \mathbf{f}_N(\omega_k) \mathbf{f}_N^H(\omega_k) \quad (3)$$

is the noise and IAA interference (signals at frequency grid points bar ω_k) covariance matrix for the k th grid point. The signal power is denoted by $p_k = |\alpha_k|^2$ and the IAA covariance matrix is given by:

$$\mathbf{R}_N = \mathbf{F}_{N,K} \mathbf{P}_K \mathbf{F}_{N,K}^H \quad (4)$$

with \mathbf{P}_K a diagonal matrix with p_k on its main diagonal. Minimisation of Eq. 2 for α_k (with p_k kept constant) $k =$

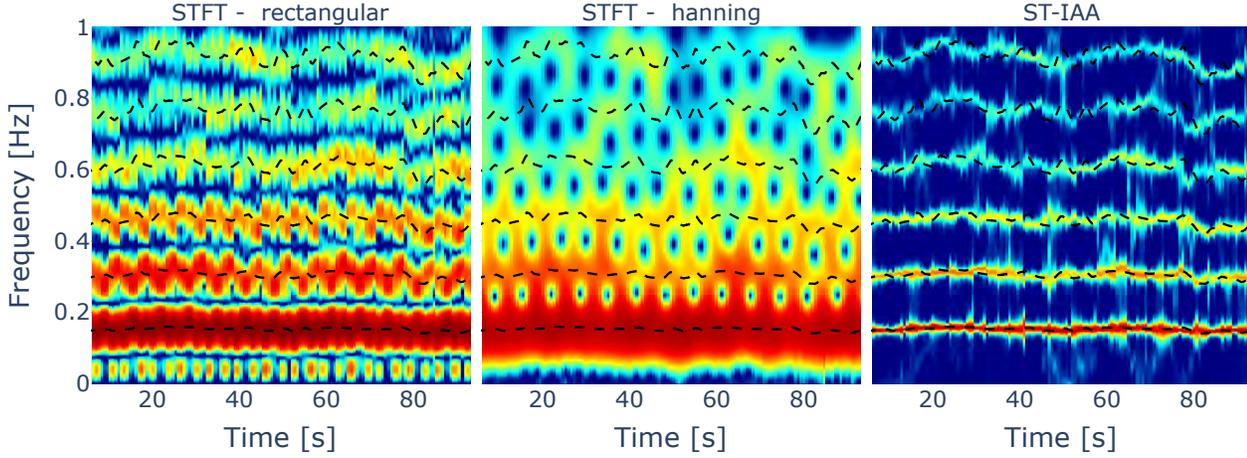


Figure 1. (Left) TFR using the STFT with a rectangular window, (Center) TFR using the STFT with a Hanning window, (Right) TFR using the ST-IAA. The black dashed lines represent the true input frequency of the simulated harmonics.

0, 1, ..., $K - 1$ results in (see Appendix A for the derivation):

$$\alpha_k^{IAA} = \frac{\mathbf{f}_N^H(\omega_k) \mathbf{Q}_N^{-1}(\omega_k) \mathbf{y}_N}{\mathbf{f}_N^H(\omega_k) \mathbf{Q}_N^{-1}(\omega_k) \mathbf{f}_N(\omega_k)}, \quad k = 0, 1, \dots, K - 1. \quad (5)$$

This can be simplified using Eq. 3 and the matrix inversion lemma (Horn & Johnson, 1985; Yardibi et al., 2010) to:

$$\alpha_k^{IAA} = \frac{\mathbf{f}_N^H(\omega_k) \mathbf{R}_N^{-1} \mathbf{y}_N}{\mathbf{f}_N^H(\omega_k) \mathbf{R}_N^{-1} \mathbf{f}_N(\omega_k)}, \quad k = 0, 1, \dots, K - 1. \quad (6)$$

Equation 6 reduces the computational cost drastically since it does not necessitate the computation of $\mathbf{Q}_N^{-1}(\omega_k)$ for each frequency bin k .

Since the signal power P_K is required in Eq. 6, the IAA estimate needs to be computed iteratively. In this paper, the periodogram is used to initialize the IAA estimate. To speed up computations, the fast implementation of the IAA using the Gohberg-Semencul representations and trigonometric polynomials is employed, for more details see (Glentis & Jakobsson, 2011).

To get the short-time IAA (ST-IAA) time-frequency representation of the vibration signal, a sliding window approach is used similar to the STFT. In case minimal computation time is crucial, further optimizations can be made by approximating the ST-IAA, e.g. by assuming the covariance matrix does not change drastically from one window to the next (i.e. for window i $\mathbf{R}_N^i \approx \mathbf{R}_N^{i-1}$), or by incorporating a single step steepest descent scheme instead of using the Levinson-Durbin algorithm in the efficient formulation of the IAA (for more details, see (Glentis & Jakobsson, 2010)). In this paper, the standard ST-IAA is used without the two mentioned optimizations.

3. SIMULATION RESULTS

To analyse the performance of the ST-IAA and compare it with the STFT, a non-stationary vibration signal is simulated that is representative of a slow rotating machine. The vibration $x(n)$ with a sample period of T consists of multiple harmonics with additive white Gaussian noise ν and is described by following signal model:

$$x(n) = \sum_{m=1}^M A_m \sin(2\pi T \sum_{n=0}^{N-1} f_m(n)) + \nu(n) \quad (7)$$

where $m = 1, 2, \dots, M$ is the harmonic number, A_m is the amplitude of harmonic m , $n = 0, 1, \dots, N - 1$ is the sample number, and $f_m(n)$ is the varying frequency vector of harmonic m .

The simulated signal is 100 seconds long and consists of 6 harmonics of a fundamental frequency at 0.15 Hz that varies randomly but smoothly around this base frequency. The amplitudes decrease inversely with the harmonic number. The same input parameters are used for the STFT as for the ST-IAA, viz. a window length of 200 samples, an overlap of 95%, and a grid size of 8000 samples. Figure 1 shows the TFRs for the STFT using both a rectangular and hanning window next to the TFR using the ST-IAA.

As can be seen from Fig. 1, the ST-IAA produces a far clearer TFR to interpret due to the much narrower peaks and reduced sidelobe levels. The normalized Renyi entropy, which has often been employed in the past when measuring TFR complexity (Flandrin, Baraniuk, & Michel, 1994; Baraniuk, Flandrin, Janssen, & Michel, 2001; Susic, Saulig, & Boashash, 2011;

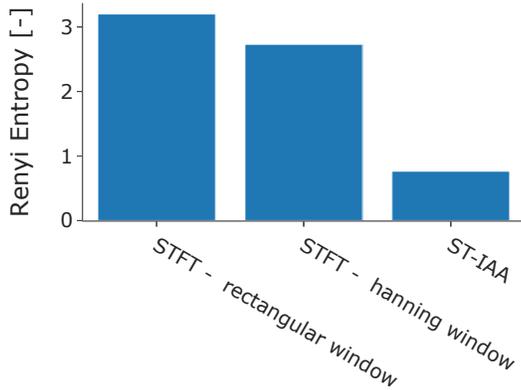


Figure 2. Renyi entropies for the three investigated TFRs shown in Fig. 1 with $\alpha = 3$.

Colominas, Jomaa, Jrad, Humeau-Heurtier, & Van Bogaert, 2017), is used to compare the complexity of each TFR. The normalized Renyi entropy of order α for a discrete-time TFR $P_x(n, k)$ is defined as follows:

$$R_\alpha = \frac{1}{1 - \alpha} \log_2 \left(\frac{\sum_{n=0}^N \sum_{k=0}^K P_x^\alpha(n, k)}{\sum_{n=0}^N \sum_{k=0}^K P_x(n, k)} \right) \quad (8)$$

where n is the discrete time variable and k the frequency bin. The third-order ($\alpha = 3$) Renyi entropy is chosen for the comparison as it was found to be most suitable for the intended purpose of TFR complexity quantification (Williams, Brown, & Hero III, 1991; Flandrin et al., 1994; Williams, 1996; Baraniuk et al., 2001). The Renyi entropies for the three TFRs of Fig. 1 are shown in Fig. 2. The ST-IAA exhibits a considerably lower Renyi entropy when compared to the STFT. While Renyi entropy is not a perfect measure to quantify the accuracy and robustness of a TFR, it does corroborate the visual interpretation of Fig. 1.

To further illustrate the improved resolution of the ST-IAA as compared to the STFT, a simple maximum tracking is done over time for the different harmonics. This is a straightforward approach to estimate rotation speeds of a machine without needing much processing power or signal processing know-how and is thus often utilized in industry. Figure 3 shows the estimated curves for each of the harmonics using the same input parameters for all three TFRs of Fig. 1. As can be seen visually, the tracking error is considerably lower for the ST-IAA as compared to the STFT. This observation is quantified by the mean square error as displayed in Fig. 4.

4. EXPERIMENTAL RESULTS

To evaluate the utility of the short-time IAA in real-world scenarios, it is compared to the STFT with a rectangular window on a vibration data set measured on the drivetrain of an offshore multi-megawatt wind turbine. Accelerometers were installed spread out over the drivetrain together with a single-pulse-per-revolution angle encoder on the high-speed shaft of the gearbox. All measurements were acquired at a sample rate of 20kHz for a duration of 10 seconds. The speed estimation from the angle encoder provides a means to assess the accuracy of the ST-IAA for tracking the speed-dependent harmonics generated by the mechanical components on top of the visual improvement in the time-frequency representations that enhance its interpretability.

Figure 5 displays the TFRs of the STFT and ST-IAA for a vibration signal measured on the first planetary gearbox stage. The used input parameters are identical for both TFRs. A window of 4 seconds is used with an overlap of 99%. Each windowed signal is also zero-padded till 40 seconds, i.e., a zero-padding factor of 10. The TFRs are zoomed in on the low frequency range from 0 to 1 Hz. Typically, the 3P frequency (i.e., three times the rotor speed) forms a distinct signature in this sub-1Hz-region for a three-bladed rotor. This is exactly what is visible in Fig. 5 around 0.68 Hz. There is another lower frequency harmonic present but this is an interfering non-speed related harmonic from an adjacent component, that also coincides partly with the first side-side natural frequency of the tower. Unfortunately the measurement duration of 10 seconds is too short to clearly distinguish between these two components.

To quantify this perceived accuracy of the ST-IAA in Fig 5 for post-processing techniques such as maximum tracking, the mean and median absolute errors are shown for both the STFT and ST-IAA in Fig. 6. As can be observed from Fig. 6, the errors are lower for ST-IAA as compared to the STFT, indicating that the ST-IAA is also at least as reliable as the STFT with regard to accuracy even for experimental vibration signals.

To further illustrate the potential of the ST-IAA for the analysis of noisy vibration signals, Figure 7 shows the STFT and ST-IAA TFRs for the high-speed stage sensor zoomed around the rotational speed of the high-speed shaft. The encoder speed is shown by a black full line and coincides with the high-speed shaft harmonic. However, while the high-speed shaft harmonic around 24 Hz can be observed, its SNR is considerably lower than the gear meshing frequency of the first planetary stage around 23 Hz. It is also easier to distinguish in the ST-IAA than the STFT. The same exercise in maximum tracking is done for the high-speed shaft harmonic. The estimated curve is shown in Fig. 7 by the dashed black line. The mean and median absolute errors are displayed in Fig. 8, which corroborates again the efficacy of the ST-IAA

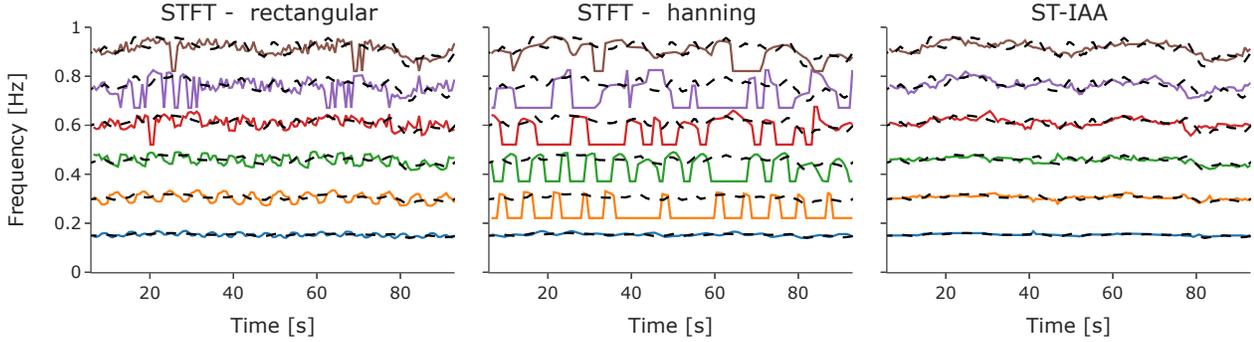


Figure 3. Estimated harmonic frequencies based on maximum tracking in a band around each harmonic for the three investigated TFRs shown in Fig. 1.

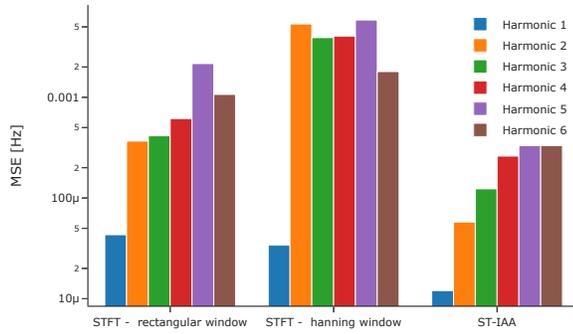


Figure 4. Mean square error of the estimated harmonic frequencies based on maximum tracking in the TFRs of Fig. 1.

in improving the STFT at the cost of additional computation time.

5. CONCLUSION

This paper investigates the potential of the short-time iterative adaptive approach (ST-IAA) as a robust and accurate non-parametric spectral estimator for time-frequency representations (TFRs). It evaluates and compares the ST-IAA to the short-time Fourier transform (STFT) on both simulated and experimental wind turbine vibration data. The ST-IAA shows that it suffers significantly less from high sidelobe levels which the STFT does suffer from and reduces both the interpretability of the time-frequency representation and its potential for post-processing techniques. For example, techniques that employ the TFR for tracking harmonic frequen-

cies can be hindered by such high sidelobe levels as is shown in both the simulation and experimental investigation. The main downside of the ST-IAA is its computation time which is considerably higher than that of the STFT. However, fast implementations of the ST-IAA do exist that alleviate some of this computational burden.

ACKNOWLEDGMENT

Cédric Peeters and Jan Helsen received funding from the Flemish Government (AI Research Program). They would like to acknowledge FWO (FondsWetenschappelijk Onderzoek) for their support through the postdoctoral grant of Cédric Peeters (1282221N). They would also like to acknowledge FWO for the support through the SBO Robustify project (S006119N).

APPENDIX A

Minimizing the weighted least-squares cost function in Eq. 2 boils down to finding α_k for which the derivative of the cost function with respect to α_k or its conjugate is zero. To simplify the notation, the dependency of f_N and Q_N^{-1} on ω_k is dropped. The derivative can then be expressed and simplified as follows:

$$\frac{\delta}{\delta \alpha_k^H} \left[(\mathbf{y}_N - \mathbf{f}_N \alpha_k)^H \mathbf{Q}_N^{-1} (\mathbf{y}_N - \mathbf{f}_N \alpha_k) \right] = 0 \quad (9)$$

$$\frac{\delta}{\delta \alpha_k^H} \left[\mathbf{y}_N^H \mathbf{Q}_N^{-1} \mathbf{y}_N - \alpha_k^H \mathbf{f}_N^H \mathbf{Q}_N^{-1} \mathbf{y}_N - \alpha_k \mathbf{y}_N^H \mathbf{Q}_N^{-1} \mathbf{f}_N + \alpha_k^H \alpha_k \mathbf{f}_N^H \mathbf{Q}_N^{-1} \mathbf{f}_N \right] = 0 \quad (10)$$

$$-\mathbf{f}_N^H \mathbf{Q}_N^{-1} \mathbf{y}_N + \alpha_k \mathbf{f}_N^H \mathbf{Q}_N^{-1} \mathbf{f}_N = 0 \quad (11)$$

$$\alpha_k^{IAA} = \frac{\mathbf{f}_N^H \mathbf{Q}_N^{-1} \mathbf{y}_N}{\mathbf{f}_N^H \mathbf{Q}_N^{-1} \mathbf{f}_N} \quad (12)$$

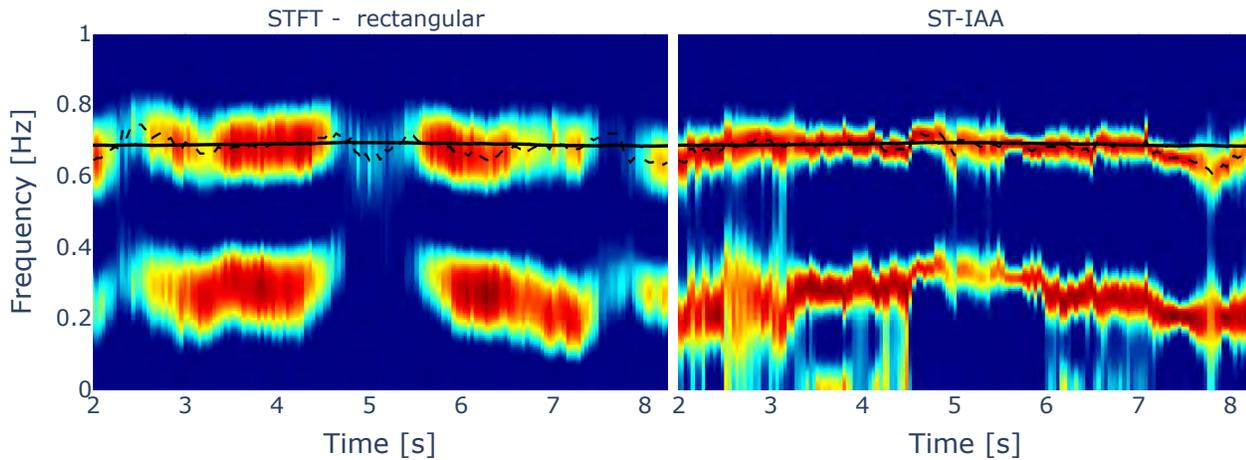


Figure 5. (Left) TFR of the vibration measured by the accelerometer installed on the first planetary gearbox stage using the STFT with a rectangular window, (Right) TFR of the same vibration but using the ST-IAA. The black full line represents the speed measured by the angle encoder while the dashed lines represent the estimated harmonic frequency based on maximum tracking.

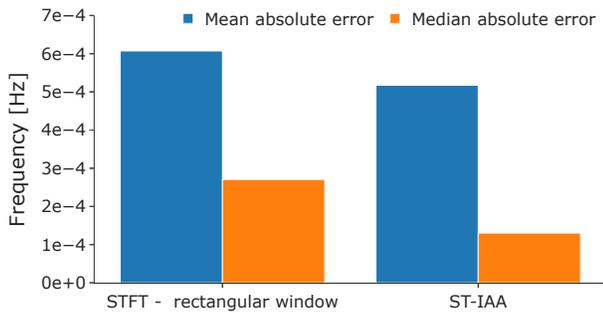


Figure 6. Mean and median absolute error between the estimated 3P harmonic frequency using maximum tracking and the encoder-based speed for the STFT (left) and the ST-IAA (right).

It can be seen that Eq. 12 is thus the same as the one in Eq. 2 for a single grid point k .

REFERENCES

- Baraniuk, R. G., Flandrin, P., Janssen, A. J., & Michel, O. J. (2001). Measuring time-frequency information content using the rényi entropies. *IEEE Transactions on Information theory*, 47(4), 1391–1409.
- Barkat, B., & Boashash, B. (1999). Design of higher order polynomial wigner-ville distributions. *IEEE Transactions on Signal Processing*, 47(9), 2608–2611.
- Baydar, N., & Ball, A. (2001). A comparative study of acoustic and vibration signals in detection of gear failures using wigner-ville distribution. *Mechanical systems and signal processing*, 15(6), 1091–1107.
- Bozkurt, B., Germanakis, I., & Stylianou, Y. (2018). A study of time-frequency features for cnn-based automatic heart sound classification for pathology detection. *Computers in biology and medicine*, 100, 132–143.
- Cazelles, B., Chavez, M., Berteaux, D., Ménard, F., Vik, J. O., Jenouvrier, S., & Stenseth, N. C. (2008). Wavelet analysis of ecological time series. *Oecologia*, 156(2), 287–304.
- Chen, X., & Feng, Z. (2021). Order spectrum analysis enhanced by surrogate test and vold-kalman filtering for rotating machinery fault diagnosis under time-varying speed conditions. *Mechanical Systems and Signal Processing*, 154, 107585.
- Choi, H.-I., & Williams, W. J. (1989). Improved time-frequency representation of multicomponent signals using exponential kernels. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(6), 862–871.
- Colominas, M. A., Jomaa, M. E. S. H., Jrad, N., Humeau-Heurtier, A., & Van Bogaert, P. (2017). Time-varying time-frequency complexity measures for epileptic eeg data analysis. *IEEE transactions on biomedical engineering*, 65(8), 1681–1688.
- Daubechies, I., Lu, J., & Wu, H.-T. (2011). Syn-

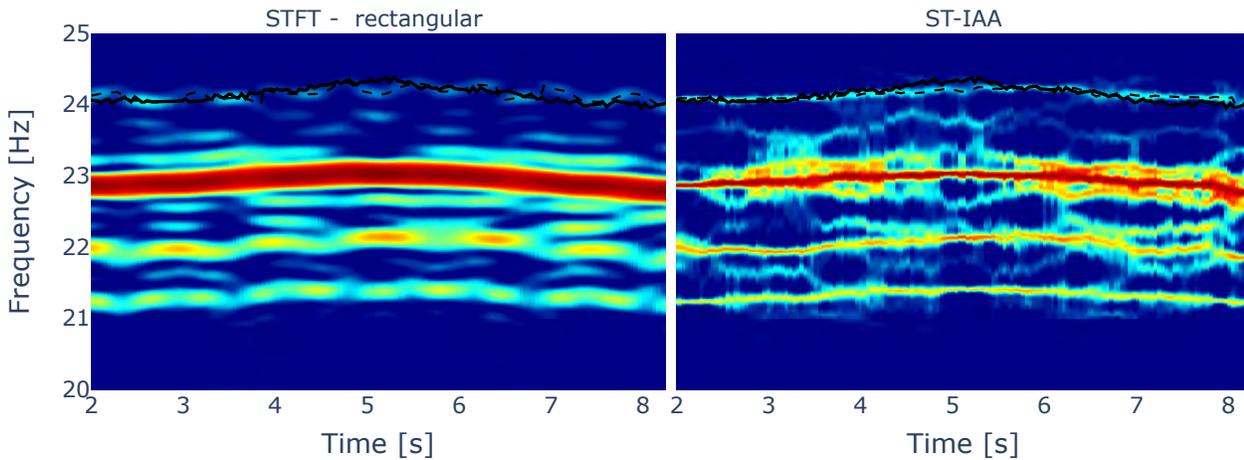


Figure 7. (Left) TFR of the vibration measured by the accelerometer installed on the high-speed gear stage using the STFT with a rectangular window, (Right) TFR of the same vibration but using the ST-IAA. The black full line represents the speed measured by the angle encoder while the dashed lines represent the estimated harmonic frequency based on maximum tracking.

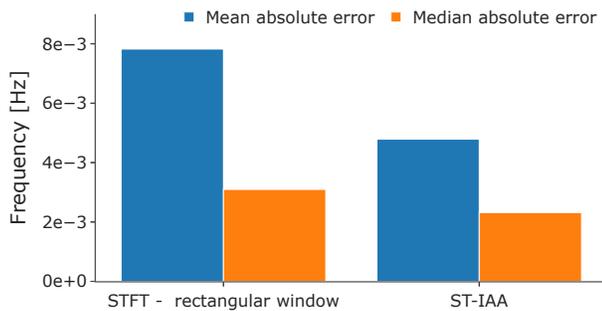


Figure 8. Mean and median absolute error between the estimated high-speed shaft harmonic frequency using maximum tracking and the encoder-based speed for the STFT (left) and the ST-IAA (right).

chrosqueezed wavelet transforms: An empirical mode decomposition-like tool. *Applied and computational harmonic analysis*, 30(2), 243–261.

Du, L., Li, J., Stoica, P., Ling, H., & Ram, S. S. (2009). Doppler spectrogram analysis of human gait via iterative adaptive approach. *Electronics letters*, 45(3), 186–188.

Feng, Z., Liang, M., & Chu, F. (2013). Recent advances in time–frequency analysis methods for machinery fault diagnosis: A review with application examples. *Mechanical Systems and Signal Processing*, 38(1), 165–205.

Flanagan, J. L., & Golden, R. (1966). Phase vocoder. *Bell System Technical Journal*, 45(9), 1493–1509.

Flandrin, P., Baraniuk, R. G., & Michel, O. (1994). Time-frequency complexity and information. In *Proceedings of icassp'94. ieee international conference on acoustics, speech and signal processing* (Vol. 3, pp. III–329).

Gabor, D. (1946). Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, 93(26), 429–441.

Gardner, T. J., & Magnasco, M. O. (2006). Sparse time-frequency representations. *Proceedings of the National Academy of Sciences*, 103(16), 6094–6099.

Glentis, G.-O., & Jakobsson, A. (2010). Time-recursive iaa spectral estimation. *IEEE Signal Processing Letters*, 18(2), 111–114.

Glentis, G.-O., & Jakobsson, A. (2011). Efficient implementation of iterative adaptive approach spectral estimation techniques. *IEEE Transactions on Signal Processing*, 59(9), 4154–4167.

Greitans, M. (2005). Adaptive stft-like time-frequency analysis from arbitrary distributed signal samples. In *International workshop on sampling theory and application, samsun, turkey*.

He, Q. (2013). Time–frequency manifold for nonlinear feature extraction in machinery fault diagnosis. *Mechanical Systems and Signal Processing*, 35(1-2), 200–218.

Horn, R., & Johnson, C. (1985). *Matrix analysis*, cambridge univ. Press. MR0832183.

Karlsson, J., Rowe, W., Xu, L., Glentis, G.-O., & Li, J. (2014). Fast missing-data iaa with application to

- notched spectrum sar. *IEEE Transactions on Aerospace and Electronic Systems*, 50(2), 959–971.
- Kravchinsky, V. A., Langereis, C. G., Walker, S. D., Dlusskiy, K. G., & White, D. (2013). Discovery of holocene millennial climate cycles in the asian continental interior: Has the sun been governing the continental climate? *Global and planetary change*, 110, 386–396.
- Labat, D. (2005). Recent advances in wavelet analyses: Part 1. a review of concepts. *Journal of Hydrology*, 314(1-4), 275–288.
- Leclere, Q., André, H., & Antoni, J. (2016). A multi-order probabilistic approach for instantaneous angular speed tracking debriefing of the cmmno 14 diagnosis contest. *Mechanical Systems and Signal Processing*, 81, 375–386.
- Liu, S., Zhang, Y. D., & Shan, T. (2018). Detection of weak astronomical signals with frequency-hopping interference suppression. *Digital Signal Processing*, 72, 1–8.
- Mann, S., & Haykin, S. (1991). The chirplet transform: A generalization of gabor's logon transform. In *Vision interface* (Vol. 91, pp. 205–212).
- Martin, N., & Mailhes, C. (2009). A non-stationary index resulting from time and frequency domains. In *Sixth international conference on condition monitoring and machinery failure prevention technologies. cm 2009 and mfpt 2009*.
- Neal, L., Briggs, F., Raich, R., & Fern, X. Z. (2011). Time-frequency segmentation of bird song in noisy acoustic environments. In *2011 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 2012–2015).
- Oberlin, T., Meignen, S., & Perrier, V. (2014). The fourier-based synchrosqueezing transform. In *2014 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 315–319).
- Peeters, C., Leclere, Q., Antoni, J., Lindahl, P., Donnal, J., Leeb, S., & Helsen, J. (2019). Review and comparison of tacholeless instantaneous speed estimation methods on experimental vibration data. *Mechanical Systems and Signal Processing*, 129, 407–436.
- Peng, Z., Li, J., Hao, H., & Xin, Y. (2020). High-resolution time-frequency representation for instantaneous frequency identification by adaptive duffing oscillator. *Structural Control and Health Monitoring*, 27(12), e2635.
- Reager, J., Thomas, B., & Famiglietti, J. (2014). River basin flood potential inferred using grace gravity observations at several months lead time. *Nature Geoscience*, 7(8), 588–592.
- Salisbury, J., & Wimbush, M. (2002). Using modern time series analysis techniques to predict enso events from the soi time series. *Nonlinear Processes in Geophysics*, 9(3/4), 341–345.
- Sapena-Bano, A., Burriel-Valencia, J., Pineda-Sanchez, M., Puche-Panadero, R., & Riera-Guasp, M. (2016). The harmonic order tracking analysis method for the fault diagnosis in induction motors under time-varying conditions. *IEEE Transactions on Energy Conversion*, 32(1), 244–256.
- Shensa, M. J., et al. (1992). The discrete wavelet transform: wedding the a trous and mallat algorithms. *IEEE Transactions on signal processing*, 40(10), 2464–2482.
- Spanos, P., Giaralis, A., & Politis, N. (2007). Time-frequency representation of earthquake accelerograms and inelastic structural response records using the adaptive chirplet decomposition and empirical mode decomposition. *Soil Dynamics and Earthquake Engineering*, 27(7), 675–689.
- Stockwell, R. G., Mansinha, L., & Lowe, R. (1996). Localization of the complex spectrum: the s transform. *IEEE transactions on signal processing*, 44(4), 998–1001.
- Sucic, V., Saulig, N., & Boashash, B. (2011). Estimating the number of components of a multicomponent non-stationary signal using the short-term time-frequency rényi entropy. *EURASIP Journal on Advances in Signal Processing*, 2011(1), 1–11.
- Torrence, C., & Compo, G. P. (1998). A practical guide to wavelet analysis. *Bulletin of the American Meteorological society*, 79(1), 61–78.
- Wang, Y. (2007). Seismic time-frequency spectral decomposition by matching pursuit. *Geophysics*, 72(1), V13–V20.
- Wang, Y., & Orchard, J. (2009). Fast discrete orthonormal stockwell transform. *SIAM Journal on Scientific Computing*, 31(5), 4000–4012.
- Wang, Y., Peter, W. T., Tang, B., Qin, Y., Deng, L., Huang, T., & Xu, G. (2019). Order spectrogram visualization for rolling bearing fault detection under speed variation conditions. *Mechanical Systems and Signal Processing*, 122, 580–596.
- Wigner, E. (1932, Jun). On the quantum correction for thermodynamic equilibrium. *Phys. Rev.*, 40, 749–759. Retrieved from <https://link.aps.org/doi/10.1103/PhysRev.40.749> doi: 10.1103/PhysRev.40.749
- Williams, W. J. (1996). Reduced interference distributions: biological applications and interpretations. *Proceedings of the IEEE*, 84(9), 1264–1280.
- Williams, W. J., Brown, M. L., & Hero III, A. O. (1991). Uncertainty, information, and time-frequency distributions. In *Advanced signal processing algorithms, architectures, and implementations ii* (Vol. 1566, pp. 144–156).
- Yardibi, T., Li, J., Stoica, P., Xue, M., & Baggeroer, A. B. (2010). Source localization and sensing: A nonparametric iterative adaptive approach based on weighted

least squares. *IEEE Transactions on Aerospace and Electronic Systems*, 46(1), 425–443.

Zhang, R., & Castagna, J. (2011). Seismic sparse-layer reflectivity inversion using basis pursuit decomposition.

Geophysics, 76(6), R147–R158.

Zheng, G., & McFadden, P. (1999). A time-frequency distribution for analysis of signals with transient components and its application to vibration analysis.

Towards data reliability based on triple redundancy and online outlier detection

Sylvain Poupry, Cédric Béler, Kamal Medjaher

Laboratoire Génie de Production, ENIT Toulouse INP, 47 Avenue d'Azereix, Tarbes, 65000, France

sylvain.poupry@enit.fr

cedrick.beler@enit.fr

kamal.medjaher@enit.fr

ABSTRACT

Today, air quality monitoring is a global concern. The World Health Organization (WHO) defined standards for each pollutant and each member state is committed to monitoring them continuously and reliably to protect the population. This responsibility is delegated to air quality monitoring associations. To achieve the objectives of reliable, accurate, and continuous measurements, these associations rely on conventional measuring stations with demanding specifications to serve as scientific references and decision supports for the authorities. However, because of heavy investments and required qualified staff, there are few stations and the coverage is coarse for territories of several thousand km². To circumvent this difficulty, measurement network architectures using Low-Cost Sensors (LCS) have been deployed. Low cost and requiring less qualification, This alternative technology to conventional measuring stations makes it possible to target local pollution that could not otherwise be detected. Although it is more accurate on the spatial dimension, this technology has several drawbacks, notably in terms of measurement repeatability and hardware quality. It is also subject to measurement drifts over time. To overcome these drawbacks, a resilient and reliable architecture based on LCS and triple redundancy has been proposed. The basic principle is based on the implementation of three smart sensors (SmS) using LCS measuring the same parameters on the same perimeter. These SmS communicate with an Aggregator that aggregates the data sent by SmS. The aggregator includes also detection and voting tasks allowing to compare, cross the data, detect faults of LCS online, and provide data that are ready for processing. In this paper, a pre-processing algorithm in four steps is presented. It identifies hardware faults from one or more LCS and reports outliers for verification by an expert. It is configurable and can identify failure behaviors (LCS or air quality). Fi-

nally, the proposed algorithm excludes the outliers data from faulty LCS and presents only reliable ones.

1. INTRODUCTION

Air pollution is the cause of 4.2 million deaths every year, not to mention the impact on wildlife. Based on this fact, air pollution is continuously monitored in a reliable and accurate way by air quality associations. As a scientific reference, these associations allow the authorities to take decisions in case of alert and to protect the population. However, these measuring stations require heavy investments and qualified personnel, and only few stations are deployed. As a consequence, the monitoring coverage is coarse and, despite extrapolations, local pollutant phenomena on territories of several km² are not detected.

In addition to the monitoring of air quality associations, deployments of measurement networks are carried out with low cost sensors (LCS) (Morawska et al., 2018). These deployments were facilitated as the LCS are inexpensive aspect (in the order of x10 to x100) and require less qualified personnel. The spatial dimension of these networks is a strong advantage, in particular for detecting local pollution and specifying the extrapolations of air quality associations (Castell et al., 2017). However, the measurements, at the level of a geographical point, present problems of precision and reliability. Indeed, LCSs have several drawbacks with respect to their material quality, measurement drift, cross-interference with other pollutants and their lifetime (Lewis et al., 2016). As a consequence, the reliability of each point of the network is questioned and the continuity of the measurement depends on the random lifetime of the LCS.

To overcome these problems, a resilient and reliable measuring station based on LCS and triple redundancy was developed. The station monitors the pollutant concentrations at a geographical point of the measurement network. It is located in a measurement perimeter where the environmental parameters do not vary at any point within the perimeter. It is com-

Sylvain Poupry et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

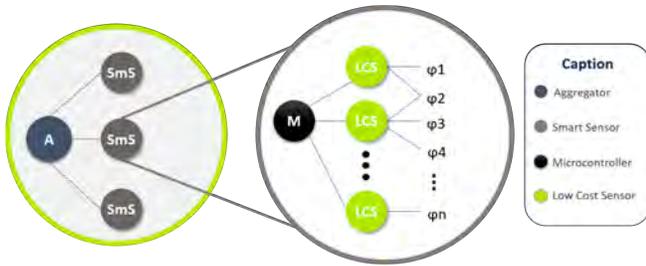


Figure 1. Measuring station composition.

posed of three (and can be extended to more) smart sensors (SmS) and an Aggregator, as shown in the figure 1.

The redundancy is active and is located at the level of the SmS. With a minimum of three, each SmS measures the same number N of parameters under the same environmental conditions. The SmS is composed of a microcontroller and the LCS measuring the N parameters (φ_i). The microcontroller concatenates and aggregates the LCS measurements into a vector φ and then communicates with the Aggregator and transmits the data at a frequency F , as shown in figure 2.

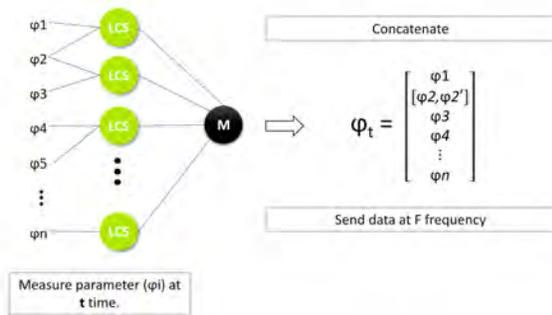


Figure 2. Processing of SmS measurements.

The Aggregator receives the vectors φ from each SmS, restructures the data by parameters (φ_i), and stores them. This first processing aggregates the raw data of the relevant LCS measurements from each SmS. The figure 3 gives an example of raw data after the restructuring step.

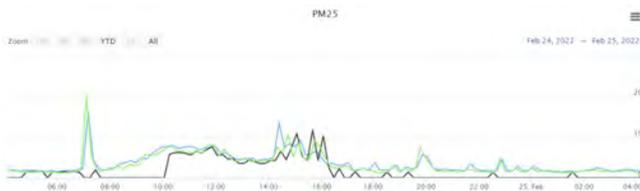


Figure 3. PM2.5 concentration measurements over one day.

However, the sending of data to the Aggregator by the SmS is not synchronized. Indeed, each SmS communicates with its own P-period making comparisons between measurements difficult. Moreover, the variable quality of the sensitive ele-

ments leads to repeatability problems. The detection of certain peaks can also be questioned. Finally, the failure of an LCS can affect the global synthesis for the targeted parameter (as shown by the black curve in figure 3). In summary, the problem of the synthesis of the measuring station is at three levels: measurement comparison, outliers detection to keep the most reliable data, and finally, the final synthesis which must be the most faithful image of the air quality measured while targeting the failing LCS or in a drift phase.

To remedy to the above mentioned issues, a method is proposed in this paper. It consists of four steps: Restructuring, Detection, Filtering, and Aggregation. It exploits the independence of the SmS and provides a global synthesis of the measurement station. An algorithm exploiting this method is implemented in the aggregator. At the input, the LCS measurements are grouped by parameters and, at the output, a synthesis exploitable in real-time is provided with the discarded data for identification of the failures. A confidence index Ic , based on the number of LCS errors, is also introduced. It is an indicator of the measuring station health state. Indeed, the station works continuously, despite the state of its components and the difficulty to maintain them because of their difficult access. The confidence index Ic is then a quick way to assess the integrity of the station operation and the reliability of the acquired data.

This paper is structured as follows: the relevant state of the art concerning detection and voting algorithms is presented in section 2, the methodology and the description of the algorithm are presented in section 3, section 4 presents the first results obtained from our case study and section 5 concludes the paper and gives some perspectives about the presented work.

2. STATE OF THE ART

The architecture of the measuring station is inspired by the triple modular redundancy (TMR) which is widely used in the industry for high availability and reliability of critical applications. The principle behind it is based on three identical and independent modules operating in parallel and having same inputs. The output of these modules is submitted to a voting unit to create an output and generate a synthesis. The aggregations performed by the voting unit are generally the majority vote, the median vote, and the weighted average vote (Lorzak, Caglayan, & Eckhardt, 1989).

The majority voting algorithms make the system fault-tolerant by selecting the output corresponding to the majority of the modules' outputs. Otherwise, the output is a safety code to safely shut down the system. The latest developments of the majority voting algorithm use the historical data to optimize the choice and thus make the system more reliable. However, this kind of majority voting algorithm has two major drawbacks. The first drawback is that when the outputs of the

modules are not close (which is the case for LCS), a threshold must be defined in order to group the close values so that the algorithm considers them identical and, when the majority is not reached, the system is stopped. The second disadvantage comes from the fact that the failures are masked. Indeed, the output corresponds to the most frequent value but the discarded values are not exploited.

Weighted average voting algorithms average the outputs of the modules with weights assigned to each output. These weights are calculated from their respective deviations. The larger the deviation, the smaller the weight. New algorithms are proposed (Latif-Shabgahi, 2004) and are more reliable according to their authors than majority voting when errors are present. Their main advantage is to provide an output whatever the number of modules present (contrary to the voting algorithms). However, when the errors are large, the number of incorrect outputs of the algorithm increases. This observation comes from the use of the average. Indeed, the mean is influenced by the extremes, and the more outliers in the inputs, the more the mean will be influenced (Leys, Ley, Klein, Bernard, & Licata, 2013). Moreover, the performance of these algorithms depends on the choice of weights.

The median voting algorithms are more efficient than those based on the weighted average. Indeed, the use of the median can allow getting rid of the influence of outliers. But the reliability of such an algorithm is diminished when the majority of the values are outliers (Bass, Latif-Shabgahi, & Bennett, 1997).

Each type of voting algorithm performs well when the input errors are a minority of the total output. They are optimized to provide an output with the least error. The combination of algorithms and the use of classification to find the best output increases the reliability but also the computational complexity and the processing time (Kassab, Hashad, Taha, & Shedied, 2013). However, for real-time processing, computation time must be taken into account. Nevertheless, whatever the combination of algorithms, they retain their weakness and remain influenced by the errors in the inputs. Finally, the common point of these algorithms is the masking of errors. This is why, in order to increase the performance of the algorithms, a step of data-driven fault detection is proposed and implemented upstream of the aggregation in the voting unit. Indeed, according to the authors of this study (Kucera, Hyncica, Cidl, & Vasatko, 2006), this configuration allows to make a TMR system more reliable.

For fault detection methods, the closest domain of the application addressed in this paper is the Wireless sensor networks (WSN). Indeed, when several stations are deployed, each station can be assimilated to a sink node where the SmS correspond to sensor nodes. The configuration is even more similar because the SmS transmit the measurements with their period P . There are various fault detection techniques and a qualita-

tive comparison of the latest fault detection algorithms for the deployment of WSN is listed in this reference (Muhammed & Shaikh, 2017). Among all possible techniques, the choice is motivated by a distributed self-fault diagnostic of sensor nodes, which are the SmS in this paper. For a large-scale deployment, each measuring station must identify its faults in order not to increase the computational complexity at the global network level. Thus, in (Panda & Khilar, 2015), an algorithm is proposed using the normalized median standard deviation to detect and discard the outliers. Consisting of two phases, the first phase of the algorithm aims to harvest measurements during an estimation time and associate them with LCS identification. The second phase discards the outliers by a statistical method using the Normalized Median Absolute Deviation (MADN). However, although the first phase is inspiring for the detection step, the calculation of MADN assumes that the distribution of the measurements is normal, which is not the case of measurements related to natural phenomena.

In conclusion, in order to make the Aggregator synthesis more reliable, it is important to incorporate a fault detection step upstream of the voting unit in order to decrease the output errors. Then, for the most reliable data provided at the output, a filtering will be applied to decrease the noise. Finally, a median voting algorithm is proposed. It is less sensitive to the extreme values compared to the average and is also more reliable than the majority voting algorithms. Therefore, the contribution of this paper consists in the development of the algorithm in four steps:

- Data restructuring of SmS vectors φ ;
- Fault Detection and storage of outliers;
- Filtering (curve smoothing);
- Aggregation by the median voter.

3. METHOD

For the input data (measurements) of the SmS, the following assumptions are made:

- Being implemented in a measurement perimeter, the LCS measure the same parameters under the same environmental conditions;
- The output data of the SmS from a parameter φ_i are the result of measurements and uncertainties of the corresponding LCS;
- The air pollution phenomenon is slow by its physical nature. Therefore, a sampling of the order of a minute would be sufficient.

In the propose method, the Aggregator receives the data from each SmS, stores them and restructures them. Then, a detection step is applied to detect faulty components and discard outliers to provide useful data. These date are then filtered and aggregated with the median voting method. As output,

we obtain a reliable synthesis, a list of errors detected with the identified components, and a confidence index correlated to the number of reliable LCS. Figure 4 summarizes the set of steps handling the input/output within the Aggregator.

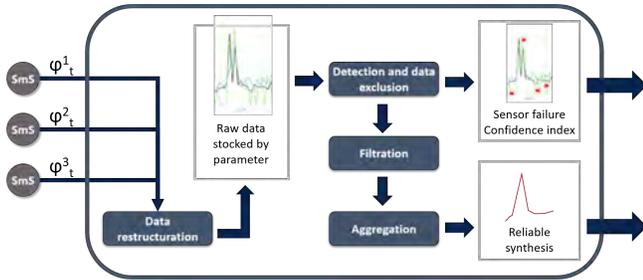


Figure 4. Aggregator's functions.

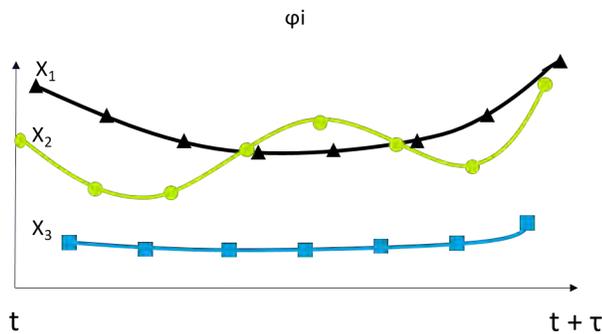
3.1. Data restructuration

Each SmS transmits to the Aggregator a measurement vector φ^j at a period P. This first step consists in reading the values of the vector and then storing them by parameter (φ_i) by associating the SmS identifier and the time of the measurement. This classification is essential to find the corresponding LCS from the SmS identification since each measured parameter is associated with a specific sensor. Thus, at the end of this restructuration step, for each SmS numbered j and for each parameter (φ_i), a time-series X_j is created, where x_h is the measurement and t the associated timestamp (Eq. (1)). The obtained time-series are then used by the fault detection step.

$$X_j = [x_{1,t}, x_{2,t+P}, \dots, x_{h,t+hP}, \dots, x_{n,t+nP}] \quad (1)$$

3.2. Fault detection step

The fault detection step applies for every parameter. It consists of two phases: initialization and detection.



$$X_j = \{x_{1,t}, x_{2,t+P}, \dots, x_{h,t+hP}, \dots, x_{n,t+nP}\} \text{ with } \tau > P \text{ and } nP < \tau$$

Figure 5. Raw data after initialisation step

The first phase consists in retrieving the measurements for an estimation time τ corresponding to the desired number of points. During initialization, τ is used to define the window size to apply a rolling window on the data set as a step-time. Its size is defined with respect to the periods of the SmS and the average time of the monitored air pollution evolution. It also must be greater than the largest value of the SmS periods P to group at least one value of each LCS. Figure 5 illustrates raw data sampling for the parameter (φ_i) with $\tau = 7P$. Due to the difference in the P period between SmS, the third SmS gives seven points instead of eight for the others.

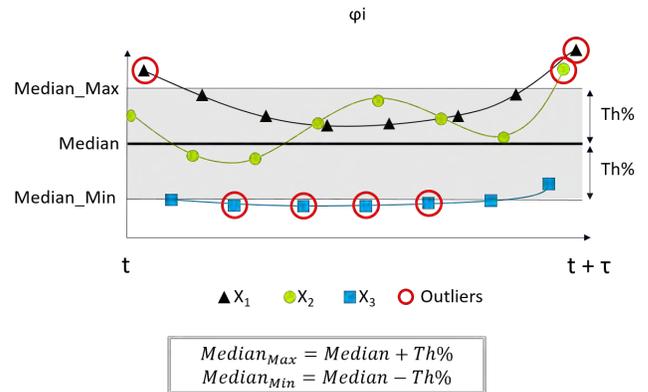


Figure 6. Raw data after initialisation step.

The second phase (detection) allows to identify the outlier data from the failed LCS. It consists, first of all, in checking the size of the time-series X_j . Indeed, missing data means that a hardware fault occurred on the associated SmS number j leading to a non-transmission of its data. The fault could be due to network failure, power failure or SmS fault. This detection triggers an alert at the output of the Aggregator which is then stored in the *Alert* series with the corresponding timestamp. The number of alerts in this series influences the confidence index which is correlated to the number of errors of LCS in service. Then, after checking the size of the times-series, the next step consists in calculating the median of all the concatenated measurement values. The median has been chosen in order to follow as closely as possible the pollution peaks, which is not the case of the mean that tends to crush them. Then a threshold deviation Th is set to calculate the $Median_{Max}$ and $Median_{Min}$ values. This deviation, defined in percentage, is estimated from the historical data. It corresponds to the maximum variability allowed between the LCS values and improves the sensitivity of the detection, as indicated in figure 6.

The values outside the area delimited by $Median_{Max}$ and $Median_{Min}$ are considered as outliers. They are stored in a time-series X_{jOut} associated to the SmS number j. The conservation of these values at the output of the Aggregator will allow to contextualize them with the following window

in order to be able to differentiate if they are anomalies due to measurements or to LCS errors. The values in the bounded area are the ones considered reliable. They are stored in a new X_{jr} time series also associated with the SmS. These data are then processed by the filtering step.

3.3. Filtering step

This step allows to attenuate the perturbations or the measurement noise specific to the LCS. The filtering using a kernel regression is applied to the X_{jr} time-series for curve smoothing. The use of Nadaraya-Watson Kernel Regression on these data is motivated by the fact that they are statistically non-parametric (measurements on environmental systems). Moreover, this approach is optimized for small numbers of points and suffers less from bias problems at the extreme points of the time-series (Nadaraya, 1964). Once the data are reliable and filtered, they are stored in a new time-series X_{jrf} associated with the corresponding SmS number and presented as inputs to the aggregation step.

3.4. Aggregation step

This step consists in applying a median voting algorithm on the reliable and filtered data obtained after the four previous steps. It consists of three tasks:

1. The values of the time-series X_{jrf} are concatenated into a vector of values named S_{rf} ;
2. The values of S_{rf} are arranged in ascending order;
3. If the number of elements of S is odd, the $(n + 1)/2$ element is selected for the output. If the number of elements of S is even then the average is calculated between the $n/2$ and $(n + 1)/2$ elements.

The output of this voting algorithm is then stored in a time-series Out_{agg} with the timestamp equal to $t + \tau$, as its index. This output corresponds to the rolling window synthesis defined in the detection step. For the whole data, the Aggregator synthesis is the set of output values from the aggregation step stored in the Out_{agg} time-series.

These four subsections describe each step of the proposed method for processing SmS data concluded by the aggregation step. The measurement vectors (φ_i) constitute the inputs of the algorithm. After these steps, various outputs are produced: the time-series $Alert$ for alerts, the reliable synthesis of the data by the time-series Out_{agg} , and the time-series X_{jOut} to perform post-processing. The whole method leads to an algorithm implemented within the Aggregator which will be presented in the next subsection.

3.5. Algorithm description

The proposed Aggregator algorithm is implemented for three SmS transmitting measurement vectors φ_1 , φ_2 , and φ_3 with their own period P. For clarity of presentation, the algorithm

described hereafter will start after the data restructuring step and has as input the raw data from a single parameter φ_i .

3.5.1. Rolling windows

The *StartTime_data* and *EndTime_data* variables correspond to the start and end timestamps of the raw data. Figure 7 shows the main algorithm and, more precisely, the rolling window for the raw data. The Processing and Output steps are detailed in figure 8.

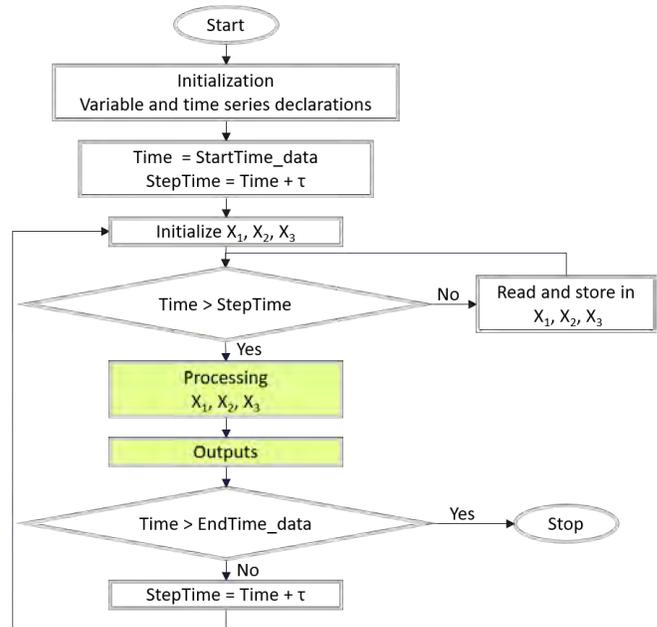


Figure 7. The main algorithm.

The initialization step consists in declaring the variables τ , Th and the time-series described in the method section. The time-series X_{jout} , $Alert$ and Out_{agg} are global and independent of the rolling window. They are used to store the Outputs when running the raw data with the Time variable. Then, the rolling window is initialized and its size is set to τ . The raw data is stored in the time-series X_1 , X_2 and X_3 as long as the Time variable is less than τ . When the variable Time reaches the size of the window or the end of the raw data, the treatments on X_1 , X_2 and X_3 are done and the outputs are stored. Finally, as long as the Time variable is less than the last Timestamps of the raw data, a new window is set and the times series X_1 , X_2 and X_3 are reset to store data again.

3.5.2. Processing and output

This part of the algorithm starts from the second phase of the fault detection step with the data having been stored in X1, X2 and X3. The size of the series is evaluated to detect the lack of data. The alerts are stored in the $Alert$ time-series and the number of elements present is subtracted from the total number of series (3 in this application) for the confidence

index. Then, the remaining values are compared against the thresholds calculated with the median and the deviation (Th) defined upstream. The outliers are stored in $X - jout$ corresponding to the X_j timestamp. The values having passed this first filter are considered reliable and are stored in X_{jr} . They are then smoothed by using a Nadaraya-Watson estimator and stored in X_{jrf} . Finally, the smoothed values are concatenated to evaluate the median which will be saved in Out_{agg} . The algorithm is repeated for each rolling window.

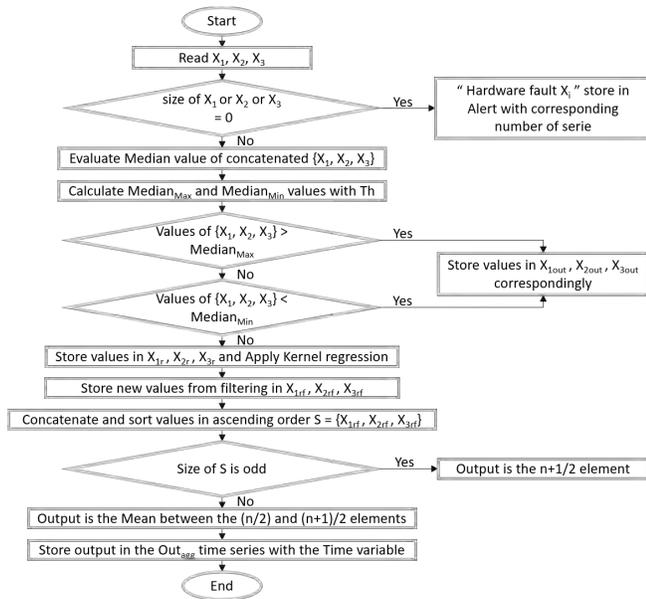


Figure 8. Processing and detailed output functions.

This subsection presents the method and the description of the algorithm. The data from the SmS are presented as input to the algorithm to be restructured by parameters. A rolling window whose size is defined upstream allows browsing the dataset. A detection module recovers the data in the first phase with a rolling window and then uses them to detect hardware faults and discard outliers in a second phase. The remaining data are filtered and presented to the voting algorithm which finalizes the synthesis for a time slot equal to the size of the rolling window.

4. RESULTS

The proposed algorithm is intended to work in real-time. To test it, the data corresponding to the measurements of PM2.5 concentration during one day are used (figure 3). The algorithm steps are programmed under Python and the code is provided in the Appendix section. The maximum period of the SmS is set to six minutes and the window size is ten minutes ($\tau = 10 \text{ min}$). The threshold value setting is defined according to the variability of the LCS observed on the data set. Initially, based on the uncertainty defined by the manufacturer, it is refined as observations are made until a thresh-

old is set. Thus, the threshold deviation Th for the detection of aberrations is fixed at 30% for the LCS in charge of the PM2.5 measurements. Indeed, the starting value chosen is 10% based on the uncertainty defined by the manufacturer then it is adjusted to be able to make the first detections. The output of the algorithm using the dataset and based on the previous settings is presented in figure 9. Figure 10 shows the algorithm output when $\tau = 20 \text{ min}$ and $Th = 30\%$. The size of the rolling windows has low impact on the number of errors collected by the algorithm. Indeed, the smaller the step size, the more peaks are included in the synthesis of the aggregator. A larger step size will flatten the synthesis curve and increase the number of outliers. The comparison of the two figures shows that when the step size is large, the peaks deviate from the synthesis but the numbers of errors on SmS 1 and 2 are approximately identical. This factor makes it possible to conclude that the anomalies are due to the measured parameter because the errors are grouped in a small time interval. The errors of SmS 3 are more frequent than those of the two others and its higher number indicates a loss of reliability of the component by its return to zero. The confidence index of this LCS can be calculated from the ratio of error to the total number of measurements. The lifting of alerts on hardware faults can also be done on the size of the time-series. Indeed, when a SmS does not send any more its data the size of its time-series will decrease compared to the other SmS.

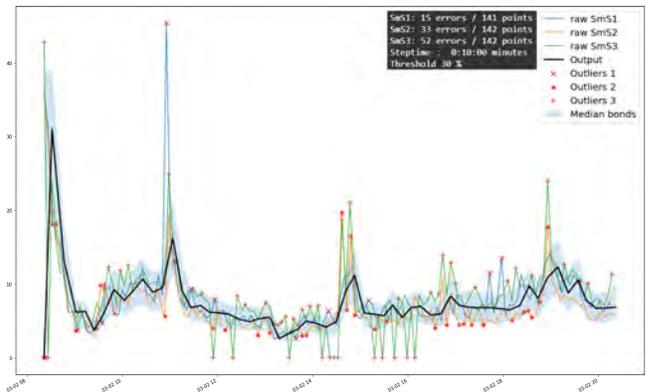


Figure 9. The algorithm output obtained on the raw data by using outliers detection with $\tau = 10 \text{ min}$ and $Th = 30\%$.

The parameter Th has a significant impact on the number of outliers. The comparisons between figures 9 and 11 and figures 10 and 12 confirm that the higher the parameter Th is, the more the deviations between measurements will be tolerated. However, this parameter can be decisive in detecting a possible offset of an LCS. Indeed, depending on the setting of Th , a high number of measurements may be detected as outliers compared to other LCS leading in a shift compared to all collected measurements.

For the synthesis of the Aggregator, the step size is an important factor. Depending on the objective of the measurement,

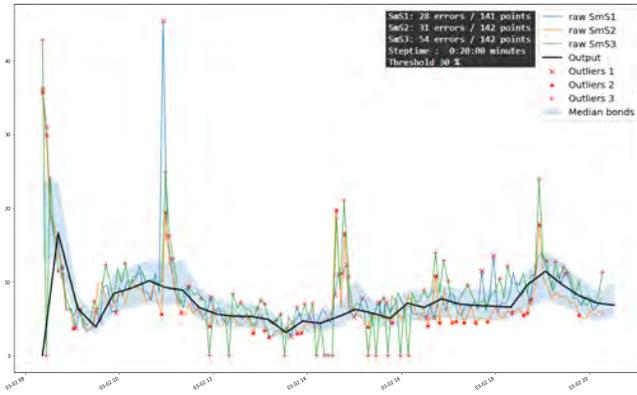


Figure 10. The algorithm output obtained on the raw data by using outliers detection with $\tau = 20 \text{ min}$ and $Th = 30\%$

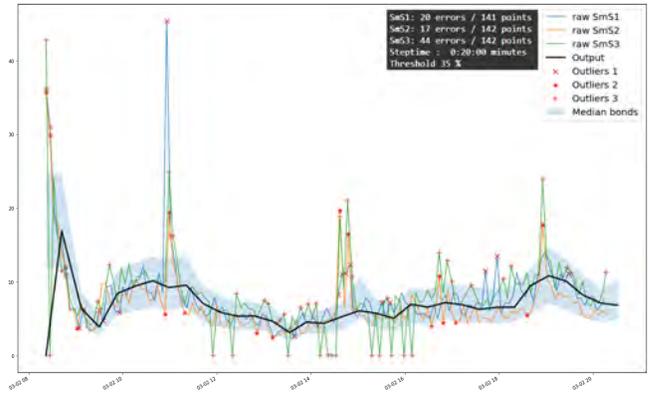


Figure 12. The algorithm output obtained on the raw data by using outliers detection with $\tau = 20 \text{ min}$ and $Th = 35\%$

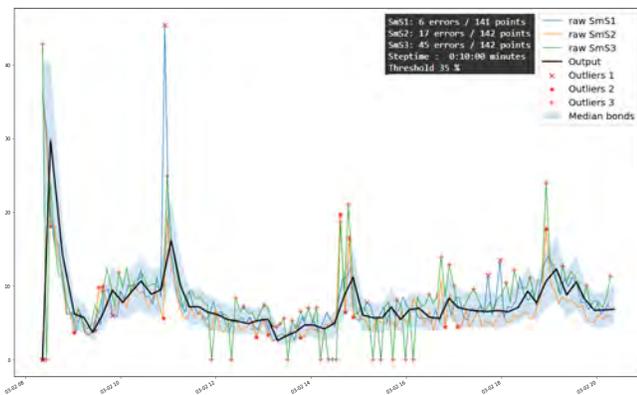


Figure 11. The algorithm output obtained on the raw data by using outliers detection with $\tau = 10 \text{ min}$ and $Th = 35\%$

it will be adapted either to avoid peak detection or to obtain a synthesis over a larger time range. This Aggregator's output is only slightly influenced by the τ parameter because of the robustness of the median on the extremes. The combination of the synthesis and the time-series outliers allows us to observe failure typologies at the hardware level but also at the air quality level. A confidence index can therefore be calculated from the number of LCS errors and the size of the input data.

5. CONCLUSION

In this paper an algorithm allowing to process pollutant concentration measurements provided by a measuring station has been proposed. It allows to have a reliable synthesis of the measurements at a given spacial point and to beyond the hardware faults of the LCS composing the SmS. The algorithm offers the possibility to adapt the purpose of measurement and the outliers detection by adjusting two parameters. The first parameter is the rolling window size τ and the second parameter is the tolerance threshold Th . The setting of Th impacts the outliers detection, which are given in a synthesis

to identify the topology of failures. Moreover, a confidence index based on the number of errors can be calculated and associated with the synthesis. Its purpose is to quickly identify the health state of the LCS in service and thus the state of the measuring station in general. Note that in this version of the algorithm, the Th and τ values are chosen empirically due to lack of long-term observation of the measuring station behavior. Therefore, as a future work, data collection through several seasons will allow to better set the detection threshold and to observe failure typologies. The stored outliers will be processed to differentiate the anomalies due to air pollution or to LCS. The classification of failures and their identification by experts will eventually allow the creation of an automatic failure identification module for preventive and predictive maintenance to be carried out on the measuring stations. The reliable data will be used to predict abnormal behaviors on a larger scale and thus to determine the air quality and eventually to predict future pollution to protect the population.

ACKNOWLEDGMENT

The authors would like to thank Rose-Marie GRENOUILLET in charge of the Environment and Serge DUTHU in charge of operations for their advice and for the setting up of the measuring station on the field.

This work is co-funded by Région Occitanie and Communauté de Communes Pyrénées Vallées des Gaves.

REFERENCES

- Bass, J., Latif-Shabgahi, G. R., & Bennett, S. (1997). *Experimental comparison of voting algorithms in cases of disagreement*. (Pages: 523) doi: 10.1109/EURMIC.1997.617368
- Castell, N., Dauge, F. R., Schneider, P., Vogt, M., Lerner, U., Fishbain, B., ... Bartonova, A. (2017, Febru-

- ary). Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates? *Environment International*, 99, 293–302. doi: 10.1016/j.envint.2016.12.007
- Kassab, M., Hashad, A., Taha, H., & Shedied, S. (2013, May). A Novel Voting Algorithm Based on Weighted Average Voting with a Classification Technique. *International Conference on Aerospace Sciences and Aviation Technology*, 15(AEROSPACE SCIENCES), 1–8. doi: 10.21608/asat.2013.22266
- Kucera, P., Hyncica, O., Cidl, J., & Vasatko, J. (2006, February). Realibility model of TMR system with fault detection. *IFAC Proceedings Volumes*, 39(21), 468–472. doi: 10.1016/S1474-6670(17)30233-1
- Latif-Shabgahi, G. R. (2004, September). A novel algorithm for weighted average voting used in fault tolerant computing systems. *Microprocessors and Microsystems*, 28(7), 357–361. doi: 10.1016/j.micpro.2004.02.006
- Lewis, A. C., Lee, J. D., Edwards, P. M., Shaw, M. D., Evans, M. J., Moller, S. J., ... White, A. (2016, July). Evaluating the performance of low cost chemical sensors for air pollution research. *Faraday Discussions*, 189(0), 85–103. (Publisher: The Royal Society of Chemistry) doi: 10.1039/C5FD00201J
- Leys, C., Ley, C., Klein, O., Bernard, P., & Licata, L. (2013, July). Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4), 764–766. doi: 10.1016/j.jesp.2013.03.013
- Lorzak, P., Caglayan, A., & Eckhardt, D. (1989, June). A theoretical investigation of generalized voters for redundant systems. In [1989] *The Nineteenth International Symposium on Fault-Tolerant Computing. Digest of Papers* (pp. 444–451). doi: 10.1109/FTCS.1989.105617
- Morawska, L., Thai, P. K., Liu, X., Asumadu-Sakyi, A., Ayoko, G., Bartonova, A., ... Williams, R. (2018, July). Applications of low-cost sensing technologies for air quality monitoring and exposure assessment: How far have they gone? *Environment International*, 116, 286–299. doi: 10.1016/j.envint.2018.04.018
- Muhammed, T., & Shaikh, R. A. (2017, January). An analysis of fault detection strategies in wireless sensor networks. *Journal of Network and Computer Applications*, 78, 267–287. doi: 10.1016/j.jnca.2016.10.019
- Nadaraya, E. A. (1964, January). On Estimating Regression. *Theory of Probability & Its Applications*, 9(1), 141–142. (Publisher: Society for Industrial and Applied Mathematics) doi: 10.1137/1109020
- Panda, M., & Khilar, P. M. (2015, February). Distributed self fault diagnosis algorithm for large scale wireless sensor networks using modified three sigma edit test. *Ad Hoc Networks*, 25, 170–184. doi: 10.1016/j.adhoc.2014.10.006

BIOGRAPHIES

Sylvain Poupry was born in France in 1986. He received a Generalist Engineering Degree with Integrated Systems Design Option at Tarbes National School of Engineering (ENIT), in July 2019. From 2007 to 2013, he was a marine officer in energy and propulsion systems in "Marine Nationale", at Toulon, France. From 2013 to 2017 he was Automation Technician and Team Leader at "Societe Financiere Altela", in Semeac, France. In 2019, he made a reconversion to be an engineer, and currently, he is a second-year doctoral student within the Production Engineering Laboratory (LGP). The topic of his Ph.D. research is "Contribution to the design and implementation of a reflexive Cyber-Physical System: application to air quality prediction in the "vallées des gaves"". His current research interests are low-cost sensors, air pollution sensing, the internet of things, data science, and Prognostics and Health Management.

Cédric Béler is assistant professor at ENIT (Ecole Nationale d'ingénieurs de Tarbes). His researches are conducted in the field of Social-Cyber-Physical System and Digital Twin and related to data science, knowledge management. He is especially interested in the way information is organized in distributed networks of information systems with human in the loop. Application are developed in the context of industry 4.0 but also in the public space (local, regional and national authorities).

Kamal Medjaher received the Engineering degree in electronics from Mouloud Mammeri University, Tizi Ouzou, Algeria, the M.S. degree in control and industrial computing from Ecole Centrale de Lille, Villeneuve-d'Ascq, France, in 2002, and the Ph.D. degree in control and industrial computing from University of Lille 1, Villeneuve-d'Ascq, France, in 2005. He was Associate Professor at the National Institute of Mechanics and Microtechnologies, Besançon, France, and FEMTO-ST Institute, from 2006 to 2016. He is currently Full Professor at Tarbes National School of Engineering (ENIT), France. He conducts his research activities within the Production Engineering Laboratory. His current research interests include prognostics and health management of industrial systems and predictive maintenance.

APPENDIX

Import library

```
import pandas as pd
from pathlib import Path
import os.path
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.nonparametric.kernel_regression import KernelReg
# For parsing dates
from datetime import datetime, timedelta
from dateutil.parser import parse
```

Import data

```
[4] # Import data from GitHub
url = 'https://raw.githubusercontent.com/spoupry/spoupry/main/pm10_daily.csv'
# Read and use csv file
# Daily PM10
df = pd.read_csv(url, delimiter='\t')

# Import data from GitHub
# url = 'https://raw.githubusercontent.com/spoupry/spoupry/main/humidite.csv'
# Read and use csv file
# For french csv from online website
df = pd.read_csv(url, delimiter=";", decimal=',') # replace , by .

print(df.head(10))
df.dtypes
```

Use database and parse data as columns index

```
# Parse date
df["DateTime"] = pd.to_datetime(df["DateTime"])
# df.head(2)
# Initialization dataset
# Setting the timestamp as dataframe's index.
df.set_index("DateTime", inplace=True) # DateTime is now the index.
print(df.head(10))
df.dtypes
```

Initialisation

```
[6] # Initialization of the variables
StartTime_df = df.index[0] # Starting timestamp dataset
EndTime_df = df.index[-1] # End timestamp dataset
tau = timedelta(minutes=10) # step size for rolling window
StepTime = StartTime_df + tau # Time slice
Threshold = 35 # 35% bonds for outliers
# Time-series for raw datas for graphics
X1raw = pd.Series([0], index = [StartTime_df], dtype=float)
X2raw = pd.Series([0], index = [StartTime_df], dtype=float)
X3raw = pd.Series([0], index = [StartTime_df], dtype=float)
# print("Start df",StartTime_df,"end df",EndTime_df,"step",StepTime)
# Initialise global time-series (4)
# Median for row datas
Median=pd.DataFrame(index=[StartTime_df],columns=['Median','MedMax','MedMin'])
# Output time-series after median voter
Outagg = pd.Series([0], index = [StartTime_df], dtype=float)
# Alert time-series from no data detection
Alert = pd.Series(['Begin'], index = [StartTime_df])
# Outliers time-series of corresponding sms
X1out = pd.Series([0], index = [StartTime_df], dtype=float)
X2out = pd.Series([0], index = [StartTime_df], dtype=float)
X3out = pd.Series([0], index = [StartTime_df], dtype=float)
# Reliable time-series extracted from Median
X1r = pd.Series([0], index = [StartTime_df], dtype=float)
X2r = pd.Series([0], index = [StartTime_df], dtype=float)
X3r = pd.Series([0], index = [StartTime_df], dtype=float)
```

Functions

```
[ ] # Calculate the median of a time-series
# Input : time-series
# Output the median as a float or None if list =[]
def calculate_median(l):
    l = sorted(l)
    l_len = len(l)
    if l_len < 1:
        return None
    if l_len % 2 == 0 :
        return ( l[(l_len)//2] + l[(l_len//2)-1] ) / 2.0
    else:
        return l[(l_len-1)//2]
```

```
[ ] # Detect Hardware fault
# Input: 3 time-series of a slice
# No Output
# Store data in Alert global time-series
def detect_no_data(X1,X2,X3):
    # Detection of no data
    # print("Start detection hardware fault")
    HardwareFault=[]
    # print("size of time-series X1,X2,X3 =",len(X1),len(X2),len(X3))
    if (len(X1) == 0):
        # print("Hardware fault SMS1")
        HardwareFault.append("Hardware Fault SMS1")
    if (len(X2) == 0):
        # print("Hardware fault SMS2")
        HardwareFault.append("Hardware Fault SMS2")
    if (len(X3) == 0):
        # print("Hardware fault SMS3")
        HardwareFault.append("Hardware Fault SMS3")
    if (len(HardwareFault) == 0):
        print("All sensors is ok")
    else:
        print(HardwareFault," Ic = ", 3 - len(Alert.loc[StepTime]),"/3")
    # Store Hardware fault in Alert time-series
    Alert.loc[StepTime] = HardwareFault
    # print("End detection hardware fault")
```

```
[ ] # Detect outlier from median and threshold
# Input: 3 time-series of a slice and a concatenated list of 3
# Output 3 time-series Xr1 of reliable values for filtering
# Store in Xr global time-series for later processing
def detect_outliers(X1,X2,X3,Concat):
    # print("Start outliers detection & store")
    # Evaluate median
    Median_slice = calculate_median(Concat)
    # Calculate MedMax and MedMin for outlier detection
    # print(calculate_median(Concat))
    MedMin = Median_slice - (Median_slice*Threshold)/100
    MedMax = Median_slice + (Median_slice*Threshold)/100
    # Store median for plotting in Median time-series
    Median.loc[StepTime,'Median'] = Median_slice
    Median.loc[StepTime,'MedMin'] = MedMin
    Median.loc[StepTime,'MedMax'] = MedMax
    # print("Median",Median_slice,"MedMin",MedMin,"MedMax",MedMax)
    # Outputs function: X reliable data time-series
    X1r1 = []
    X2r1 = []
    X3r1 = []
    idx_1r1 = []
    idx_2r1 = []
    idx_3r1 = []
    # Outliers and reliable data of X1
    if(len(X1) != 0):
        for time, row in X1.iteritems():
            if((row > MedMax) or (row < MedMin)):
                X1out.loc[time] = row
            else:
                # Store in global X1r time-series for plotting
                X1r.loc[time] = row
                X1r1.append(row)
                idx_1r1.append(time)
    # print(X1)
    # print(X1out)
    # print(X1r)
    # Outliers and reliable data of X2
    if(len(X2) != 0):
        for time, row in X2.iteritems():
            if((row > MedMax) or (row < MedMin)):
                X2out.loc[time] = row
            else:
                # Store in global X1r time-series for plotting
                X2r.loc[time] = row
                X2r1.append(row)
                idx_2r1.append(time)
    # print(X2)
    # print(X2out)
    # print(X2r)
    # Outliers and reliable data of X3
    if(len(X3) != 0):
        for time, row in X3.iteritems():
            if((row > MedMax) or (row < MedMin)):
                X3out.loc[time] = row
            else:
                # Store in global X1r time-series for plotting
                X3r.loc[time] = row
                X3r1.append(row)
                idx_3r1.append(time)
    # print(X3)
    # print(X3out)
    # print(X3r)
    # Construct time-series from lists
    X1r1 = pd.Series(X1r1, index = idx_1r1, dtype=float)
    X2r1 = pd.Series(X2r1, index = idx_2r1, dtype=float)
    X3r1 = pd.Series(X3r1, index = idx_3r1, dtype=float)
    # print("End outliers detection & store")
    return X1r1,X2r1,X3r1
```

```
[ ] # Smoothing curve by applying kernel regression
# Input: 3 time-series Xr1 of reliable values
# Output 3 time-series Xf filtered for median voter
def filtering(X1,X2,X3):
    from numpy import linspace
    # print("Start filtering")
    # X1
    # Must have two points minimum
    if(len(X1) == 0 or len(X1) == 1):
        X1f = X1
        print("not enough data")
    else:
        X1_ = X1.copy()
        X1_.index = linspace(1., len(X1_),len(X1_))
        kr = KernelReg([X1_.values],[X1_.index.values], var_type='c')
        f1 = kr.fit([X1_.index.values])
        X1f = pd.Series(data=f1[0], index = X1.index, dtype=float)
    # X2
    # Must have two points minimum
    if(len(X2) == 0 or len(X2) == 1):
        X2f = X2
        print("not enough data")
    else:
        X2_ = X2.copy()
        X2_.index = linspace(1., len(X2_),len(X2_))
        kr = KernelReg([X2_.values],[X2_.index.values], var_type='c')
        f2 = kr.fit([X2_.index.values])
        X2f = pd.Series(data=f2[0], index = X2.index, dtype=float)
    # X3
    # Must have two points minimum
    if(len(X3) == 0 or len(X3) == 1):
        X3f = X3
        print("not enough data")
    else:
        X3_ = X3.copy()
        X3_.index = linspace(1., len(X3_),len(X3_))
        kr = KernelReg([X3_.values],[X3_.index.values], var_type='c')
        f3 = kr.fit([X3_.index.values])
        X3f = pd.Series(data=f3[0], index = X3.index, dtype=float)
    # print("End filtering")
    # Return filtered data time-series
    return X1f,X2f,X3f
```

```
[ ] # Algorithm of the median voter
# Input : 3 time-series reliable and filtered
# Output the synthesis as a float or None if list =[]
def Median_voter(X1,X2,X3):
    # print("Start Median Voter")
    concatenated_series=pd.concat([X1,X2,X3])
    # print(concatenated_series)
    Output = calculate_median(concatenated_series)
    # print("Output=",Output)
    # print("End Median Voter")
    # Return synthesis of median voter
    return Output
```

```
[ ] # Create X time-series in the rolling window in a tau size from dataset
# Input: a slice of dataset
# Store in global series Outagg, Xraw
def processing(df):
    print("Start processing slice")
    #create X1,X2,X3 series
    X1 = []
    X2 = []
    X3 = []
    idx_1= [] # lists for Xi time-series creation
    idx_2= []
    idx_3= []
    for time, row in df.iterrows():
        if(pd.isna(row[df.columns[0]]) == False ): # NaN detection
            X1.append(row[df.columns[0]]) # stock in x1 list
            idx_1.append(time)
            X1raw.loc[time] = row[df.columns[0]] # input in X1raw time-series
        if(pd.isna(row[df.columns[1]]) == False ):
            X2.append(row[df.columns[1]])
            idx_2.append(time)
            X2raw.loc[time] = row[df.columns[1]] # input in X2raw time-series
        if(pd.isna(row[df.columns[2]]) == False ):
            X3.append(row[df.columns[2]])
            idx_3.append(time)
            X3raw.loc[time] = row[df.columns[2]] # input in X3raw time-series
    # print(X1,X2,X3)
    # Concatenation of all elements of the rolling windows
    Concat = X1 + X2 + X3
    # Create time series
    X1 = pd.Series(X1, index = idx_1, dtype=float)
    X2 = pd.Series(X2, index = idx_2, dtype=float)
    X3 = pd.Series(X3, index = idx_3, dtype=float)
    # print("X1,X2 and X3 created")
    # Function detection hardware fault
    detect_no_data(X1,X2,X3)
    # Function detection outliers + stock values in Xout time-series
    Xreliable = detect_outliers(X1,X2,X3,Concat) # return reliable data (Xr)
    # print(Xreliable)
    # Function kernel regression return filtered data (Xrf)
    Xfiltered = filtering(Xreliable[0],Xreliable[1],Xreliable[2])
    # Function Median voter
    Output = Median_voter(Xfiltered[0],Xfiltered[1],Xfiltered[2])
    # Store ouput in the Outagg time-series
    if(Output == None):
        print("oh la la")
    else:
        Outagg.loc[StepTime] = Output
    #return X1,X2,X3
```

Main algorithm

First slice of database

```
[ ] # Initialise the first slice by reading first Timestamps from dataset
# Create the first slice of dataset for processing
# First slice of time from df dataframe
StepTime = StartTime_df + tau
Time_slice_list=[StartTime_df ]
# print("Start: ", StartTime_df, " , End: ", StepTime)
# Identificate the first slice
df_slice = df.loc[StartTime_df:StepTime].copy()
# Processing slice
processing(df_slice)
Median.loc[StartTime_df] = Median.loc[StepTime]
# print(Median)
```

Others slices until the end of database

```
[ ] # Rolling window through the dataset
# Other slices
for time, row in df.iterrows():
    if(time <= StepTime):
        This_time = time
        #print(time)
    if(time > StepTime):
        Start_step = time
        StepTime = time + tau
        Time_slice_list.append(Start_step)
        # Time_slice_list.append(StepTime)
        # print("Start: ", Start_step, " , End: ", StepTime)
        df_slice = df.loc[Start_step:StepTime].copy()
        processing(df_slice)
#Time_slice_list
```

Outputs and graphics

```
[ ] # Sizes of series for confidence index
print('SMS1:',len(X1out),'errors /',len(X1raw),'points')
print('SMS2:',len(X2out),'errors /',len(X2raw),'points')
print('SMS3:',len(X3out),'errors /',len(X3raw),'points')
print('StepTime : ', tau , 'minutes')
print('Threshold, Threshold , 'X')
# Plotting time-series

plt.figure(figsize=(24,16))
plt.scatter(X1out.index, X1out.values,label='Outliers 1', s=100, c='red' ,marker = 'x')
plt.scatter(X2out.index, X2out.values,label='Outliers 2', s=100, c='red' ,marker = '+')
plt.scatter(X3out.index, X3out.values,label='Outliers 3', s=100, c='red' ,marker = '*')
X1raw.plot(label='raw SMS1')
X2raw.plot(label='raw SMS2')
X3raw.plot(label='raw SMS3')
plt.fill_between(Median.index,Median['MedMax'],Median['MedMin'],alpha = 0.2, label='Median bonds')
Outagg.plot(label='Output', linewidth = 3, c='black' )
# t=10
# plt.title('Th=50 and tau=xi' %t)
plt.legend(fontsize=20)
plt.show()
```

Expert Knowledge Induced Logic Tensor Networks: A Bearing Fault Diagnosis Case Study

Maximilian-Peter Radtke¹, Jürgen Bock²

^{1,2} *Technische Hochschule Ingolstadt, Ingolstadt, Germany*

maximilian-peter.radtke@thi.de

juergen.bock@thi.de

ABSTRACT

In the recent past deep learning approaches have achieved some remarkable results in the area of fault diagnostics and anomaly detection. Nevertheless, these algorithms rely on large amounts of data, which is often not available, and produce outputs, which are hard to interpret. These deficiencies make real life applications difficult. Before the broad success of deep learning machine faults were often classified using domain expert knowledge based on experience and physical models. In comparison, these approaches only require small amounts of data and produce highly interpretable results. On the downside, however, they struggle to predict unexpected patterns hidden in data. Merging these two concepts promises to increase accuracy, robustness and interpretability of models. In this paper we present a hybrid approach to combine expert knowledge with deep learning and evaluate it on rolling element bearing fault detection. First, we create a knowledge base for fault classification derived from the expected physical attributes of different faults in the envelope spectrum of vibration signals. This knowledge is used to derive a similarity function for comparing input signals to expected faulty signals. Afterwards, the similarity measure is incorporated into different neural networks using a Logic Tensor Network (LTN). This enables logical reasoning in the loss function, in which we aim to mimic the decision process of an expert analyzing the input data. Further, we extend LTNs by weight schedules for axiom groups. We show that our approach outperforms the baseline models on two bearing fault data sets with different attributes and directly gives a better understanding of whether or not fault signals are influenced by other effects or behave as expected.

1. INTRODUCTION

The increased connectivity of machines and whole production halls enabled by the Industrial Internet of Things (IIoT)

Maximilian-Peter Radtke et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

has made gathering data and monitoring machines easier. We can collect condition monitoring data about machines, which in turn make new maintenance strategies possible. Before, machines were often run until they failed or maintained on a regular predefined schedule. Now, predictive maintenance, where maintenance activities are based on the remaining useful life of a machine, is widely possible. In comparison to traditional approaches this can reduce machine down times, prolong machinery lifetime and reduce resource consumption (Selcuk, 2017).

To make this possible we must diagnose and predict faults correctly based on available data. In the recent past deep learning (DL) in particular has shown great success in doing so (Fink et al., 2020; Liu, Yang, Zio, & Chen, 2018; Arinez, Chang, Gao, Xu, & Zhang, 2020; Zhao et al., 2019). Despite its broad success DL has a number of known issues. One of the major drawbacks is the necessity of large amounts of data for training DL algorithms, without which generalization is not possible (Fink et al., 2020). Even though IIoT can provide us with plenty of data, the nature of the required data often makes collection very costly or even prohibitive. For example, to gather fault data, on which diagnostics and prognostics can be performed, a machine needs to suffer that fault. Especially for large special machinery this can be very expensive and interrupt the production flow. Another disadvantage is the difficulty with which DL output can be explained (Arrieta et al., 2020). However, this explainability is essential for building trust in the results, which can only be achieved by understanding their occurrence.

According to different authors these shortcoming could be reduced by combining purely data driven models with domain knowledge and symbolic AI techniques to hybrid models (Fink et al., 2020; Arinez et al., 2020; Garcez & Lamb, 2020; Marcus, 2020). To achieve this multiple approaches have been studied in the domain of fault diagnostics and prognostics. For instance, the combination has shown to increase explainability and enable root cause analysis by taking into account input from human experts (Steenwinckel et al., 2021).

Moreover, it has shown to make models more robust and decrease the amount of “real” data needed through high-quality simulation of data based on expert knowledge (Wang, Taal, & Fink, 2021), and increase overall performance by extending the feature space with physics inspired features (Chao, Kulkarni, Goebel, & Fink, 2022).

These approaches all have in common that the knowledge and data based parts can be clearly distinguished from one another. A different way of creating hybrid models is when both approaches are completely integrated into one another. One such idea is neurosymbolic AI, where traditional rule-based AI is merged with recent developments in DL (Garcez & Lamb, 2020). Logic Tensor Networks (LTN) (Badreddine, Garcez, Serafini, & Spranger, 2022), which use a logic language to create a loss function for training arbitrary neural networks, are a realization of this. This approach has shown to be effective on different tasks like semantic image interpretation (Donadello, Serafini, & Garcez, 2017) or deductive reasoning (Bianchi & Hitzler, 2019). To the best of our knowledge, this method has not yet been applied for classifying faults.

Our contribution is to apply the neurosymbolic framework LTN to the fault diagnostics domain. This is demonstrated by showing its suitability for incorporating knowledge into DL in the special case of bearing fault diagnostics with constant shaft speed. For this, we propose a new scoring function based on physical knowledge for classifying bearing faults. The scoring function acts as domain expertise for the LTN loss function, which can thereby be injected into classic DL models. Additionally, we extend the LTN framework by assigning weights to different axiom groups and propose weight schedules along the training process for doing so. We show that our approach outperforms the baseline methods in terms of test accuracy.

In the remainder of this paper we will first lay the foundations for Logic Tensor Networks in Sec. 2. Next, in Sec. 3 our approach for combining knowledge with DL based on LTNs for bearing fault classification is outlined. Finally, we present our experimental setup in Sec. 4 and evaluate our approach on two well known bearing fault data sets in Sec. 5.

2. FOUNDATIONS

A Logic Tensor Network (LTN) is a neurosymbolic framework, which uses a fully differentiable first-order logic language called Real Logic in the loss function for training a neural network (NN). Here we give an overview of Real Logic and how LTNs learn. For an in depth treatment of these topics we refer to the original work (Badreddine et al., 2022).

Real Logic is defined on a first-order logic language $\mathcal{L} = (\mathcal{C}, \mathcal{P}, \mathcal{F}, \mathcal{X})$, where \mathcal{C} is a set of constant symbols, \mathcal{P} a set of relational symbols (predicates), \mathcal{F} a set of functional symbols

and \mathcal{X} a set of variable symbols. \mathcal{L} allows for defining logical formulas, e.g., $\forall x((x > \text{thr}) \rightarrow \text{isFault}(x, c))$, which states that all x , that are greater than thr , yield the fault c .

Due to the desired applicability to real world problems fuzzy semantics is used, meaning that the truth value of a logical formula is between 0 and 1 in comparison to just being *true* or *false*. One of the main benefits of Real Logic is its differentiability, which is achieved by grounding onto the real plane, i.e., all logical expressions are mapped onto \mathbb{R} . For this to work all elements of \mathcal{L} are typed and attributed to a domain, e.g. the constant *Ingolstadt* is of the domain *City*.

The functions \mathbf{D} , \mathbf{D}_{in} and \mathbf{D}_{out} return the domains of the elements of \mathcal{L} and are defined as

$$\begin{aligned} \mathbf{D} &: \mathcal{X} \cup \mathcal{C} \mapsto \mathcal{D}, \\ \mathbf{D}_{\text{in}} &: \mathcal{F} \cup \mathcal{P} \mapsto \mathcal{D}^*, \\ \mathbf{D}_{\text{out}} &: \mathcal{F} \mapsto \mathcal{D}, \end{aligned}$$

where \mathcal{D} is a non-empty set of symbols called domain symbols and \mathcal{D}^* is the Kleene Star of \mathcal{D} , which is defined as the set of all finite sequences of symbols in \mathcal{D} . Thus, \mathbf{D} outputs the domain of a constant or variable, \mathbf{D}_{in} gives us the domain of the input for a function or predicate and \mathbf{D}_{out} returns the output domain for functions.

Grounding means that domains are interpreted concretely as tensors in the real field, constants and variables as tensors of real values, functions as real functions or tensor operations and predicates as functions or tensor mappings to a value in the interval $[0, 1]$. Formally, a grounding \mathcal{G} satisfies the conditions

$$\begin{aligned} \forall x \in \mathcal{X} \cup \mathcal{C} : \mathcal{G}(x) &\in \prod_{i=1}^k \mathcal{G}(\mathbf{D}(x)), \\ \forall f \in \mathcal{F} : \mathcal{G}(f) &\in \mathcal{G}(\mathbf{D}_{\text{in}}(f)) \mapsto \mathcal{G}(\mathbf{D}_{\text{out}}(f)), \\ \forall p \in \mathcal{P} : \mathcal{G}(p) &\in \mathcal{G}(\mathbf{D}_{\text{in}}(p)) \mapsto [0, 1], \end{aligned}$$

with k being the number of instances of x . A grounding depending on a set of parameters θ is depicted as $\mathcal{G}(\cdot|\theta)$. This definition can be expanded to also include all first-order terms and atomic formulas by consecutive application of the grounding function. See (Badreddine et al., 2022) for details.

To ground a complete atomic formula connectives and quantifiers are necessary. We define these in accordance to Product Real Logic as proposed by (Badreddine et al., 2022). Hence, connectives, i.e., conjunction (\wedge), disjunction (\vee), implication (\rightarrow) and negation (\neg), are defined on first-order fuzzy logic semantics (Hájek, 2013) and are associated with a t-norm (T), t-conorm (S), fuzzy implication (I) or fuzzy nega-

tion (N) respectively. These are defined as

$$\begin{aligned} T(a, b) &= ab, \\ S(a, b) &= a + b - ab, \\ I(a, b) &= 1 - a + ab, \\ N(a) &= 1 - a, \end{aligned}$$

with fuzzy logic values $a, b \in [0, 1]$. Additionally, quantifiers are defined via an aggregation operator $A : \bigcup_{n \in \mathbb{N}} [0, 1]^n \mapsto [0, 1]$. The for-all (\forall) quantifier is defined as the p-mean error

$$A_{\text{pME}}(a_1, \dots, a_n) = 1 - \left(\frac{1}{n} \sum_{i=1}^n (1 - a_i)^p \right)^{1/p}, \quad (1)$$

and exists (\exists) as the p-mean

$$A_{\text{pM}}(a_1, \dots, a_n) = \left(\frac{1}{n} \sum_{i=1}^n a_i^p \right)^{1/p},$$

given the fuzzy truth values a_1, \dots, a_n and $p \geq 1$.

Using these definitions we can now ground an exemplary atomic formula $\phi = \forall x((x > \text{thr}) \rightarrow \text{isFault}(x, c))$ with variable x , constants thr, c and predicate $\text{isFault}(x, c)$, which depends on parameters θ , and $(x > \text{thr})$. The respective domains are given by $\mathbf{D}(\cdot)$ and $\mathbf{D}_{\text{in}}(\cdot)$. Therefore, the formula is grounded through

$$\mathcal{G}(\phi|\theta) = A_{\text{pME}}(I(\mathcal{G}((x > \text{thr})), \mathcal{G}(\text{isFault}(x, c)|\theta))).$$

Real logic can be used to create a knowledge base, which is defined by the triple $\mathcal{T} = \langle \mathcal{K}, \mathcal{G}(\cdot|\theta), \Theta \rangle$. Here \mathcal{K} is a set of closed first-order logic formulas (axioms) defined on the set of symbols $\mathcal{S} = \mathcal{C} \cup \mathcal{P} \cup \mathcal{F} \cup \mathcal{X} \cup \mathcal{D}$ and Θ is the hypothesis space for the parameters θ . The goal is to learn some set of parameters that maximizes the satisfiability of the knowledge base

$$\theta^* = \underset{\theta \in \Theta}{\text{argmax}} \underset{\phi \in \mathcal{K}}{\text{SatAgg}} \mathcal{G}(\phi|\theta), \quad (2)$$

where SatAgg is some aggregation operator. In the following we will use the p-mean error defined in Eq. (1). This formulation can then be used as the loss function of a NN, which searches for an optimal set of parameters that maximizes the satisfiability. A NN with a loss function based on Real Logic is called a LTN.

Due to the differentiability of Real Logic and therefore of the optimization problem in Eq. (2), the loss function can be applied like any other loss function to a NN and all established optimization techniques and network architectures can be used.

An implementation of LTNs in Python based on Tensorflow

is available.¹

3. APPROACH

3.1. Scoring Function for Bearing Faults

For identifying bearing faults we propose to create a heuristic scoring function, that acts similar to the way an expert would analyze a bearing vibration signal. The function is inspired by the analysis done on the CWRU bearing data set by (Smith & Randall, 2015) and how the authors achieved their assessment by relying on the expected frequencies for specific faults. Depending on the location of the fault these are called ball pass frequency, outer race (BPFO), ball pass frequency, inner race (BPFi) and ball (roller) spin frequency (BSF) and are described by

$$\text{BPFO} = \frac{n_r}{2} \left(1 - \frac{d_r}{d_p} \cos \psi \right), \quad (3)$$

$$\text{BPFi} = \frac{n_r}{2} \left(1 + \frac{d_r}{d_p} \cos \psi \right), \quad (4)$$

$$\text{BSF} = \frac{d_p}{2d_r} \left(1 - \left(\frac{d_r}{d_p} \cos \psi \right)^2 \right), \quad (5)$$

where n_r denotes the number of rolling elements, d_r the roller diameter, d_p the pitch diameter and ψ the contact angle of the bearing. These frequencies are multiplied by multiples of the shaft speed v_r to obtain the expected fault frequencies over the complete spectrum.

These frequencies are more easily identified when inspecting the envelope spectrum of the vibration signal in comparison to the spectrum of the raw signal. Therefore, we first transform the raw signal into the envelope signal using the Hilbert transform (Bonnardot, Randall, Antoni, & Guillet, 2004). Afterwards, the envelope signal is transferred to the corresponding envelope spectrum. Here the different fault types outer race fault (OR), inner race fault (IR) and ball fault (B) can be recognized by the peaks at the respective fault frequencies BPFO, BPFi and BSF (Smith & Randall, 2015). The fault frequencies are especially visible in lower frequency areas (Randall & Antoni, 2011). Therefore, we only analyze frequencies of up to 500 Hz. To ensure, that the remnants of a fractional pass of a faulty bearing does not disturb our score calculation, we additionally limit ourselves to frequencies higher than one and a half times the shaft speed. We write the operation of transforming a signal x into its envelope spectrum consisting of frequencies $h = (h_1, \dots, h_n)$ and corresponding amplitude values $d = (d_1, \dots, d_n)$, with $h_n \leq 500$ and $h_1 \geq 1.5v_r$, as $\text{EnvSpec}(x)$.

In the envelope spectrum we can identify peaks by taking the moving average along a predefined window to see which points (h_i, d_i) vary strongly from the norm. We use the well-

¹<https://github.com/logictensornetworks/logictensornetworks>

known definition of the moving average

$$\text{MA}^{(w)}(d) = \frac{1}{w} \sum_{i=0}^{w-1} d_{t-i},$$

over a window of w values. We set this window to the shaft speed v_r and omit the superscript.

Let $\mathbb{1}_{\{A\}}$ be the indicator function, which is one if A is true and zero otherwise. Then, if a frequency value h_i has a corresponding amplitude d_i of 2.5 times the moving average, we can mark these as peaks and count them with

$$N_P(d) = \sum_{i=1}^n \mathbb{1}_{\{d_i \geq 2.5 \text{MA}(d,i)\}}.$$

The identified peaks are then classified into whether or not they are expected, i.e., they are at the expected fault frequencies derived from Eqs. (3), (4) and (5). The expected fault frequencies are given by $F^{(l)} = (F_1^{(l)}, \dots, F_m^{(l)})$ for each fault type $l \in \{\text{IR}, \text{OR}, \text{B}\}$. In the following we will omit the superscript l and remark that the rest of the score calculation procedure is applied for all l independently.

Due to the imperfections of the real world we relax the condition of a peak being exactly at the expected fault frequency by the value $\delta = v_r/2$ and get the number of expected peaks

$$N_{EP}^{(j)}(h, d) = \sum_{i=1}^n \mathbb{1}_{\{d_i \geq 2.5 \text{MA}(d,i)\}} \mathbb{1}_{\{F_j - \delta \leq h_i \leq F_j + \delta\}}.$$

If multiple peaks fall into the same $F_j \pm \delta$ section the weighted average is taken by calculating the value

$$V_{EP}^{(j)}(h, d) = \sum_{i=1}^n \mathbb{1}_{\{d_i \geq 2.5 \text{MA}(d,i)\}} \mathbb{1}_{\{F_j - \delta \leq h_i \leq F_j + \delta\}} h_i,$$

and dividing it through the amount of peaks. Combining this for all expected frequencies F_1, \dots, F_m this gives us

$$\bar{V}_{EP}(h, d) = \sum_{j=1}^m V_{EP}^{(j)}(h, d) / N_{EP}^{(j)}(h, d).$$

The value of all peaks, which do not fall in the expected category, is calculated by

$$V_{PNE}^{(j)}(h, d) = \sum_{i=1}^n \mathbb{1}_{\{F_{j-1} + \delta \leq h_i \leq F_{j-1} - \delta\}} d_i,$$

with $F_0 = -\delta$ and $F_{m+1} = \text{inf}$. By combining the previous equations we obtain the score function

$$f_l(h, d) = \frac{\bar{V}_{EP}(h, d)}{\bar{V}_{EP}(h, d) + \sum_{j=1}^{m+1} V_{PNE}^{(j)}(h, d)} \quad (6)$$

for each fault type l , which returns a value between 0 and 1.

An exemplary visualization for how the score is calculated is given in Fig 1.

The value can be interpreted as the share of peaks, which were as expected for a specific fault. Therefore, we set the threshold $\text{thr}_{\text{score}} = 0.49$ for determining whether one is confident in the occurrence of the respective fault. Hence, we obtain the logical formula

$$\forall x ((f_l(\text{EnvSpec}(x)) > \text{thr}_{\text{score}}) \rightarrow P(x, c_l)), \quad (7)$$

with the respective fault label l and predicate P for fault classification. With this formula we aim to mimic the way an expert would analyze a vibration signal and identify a certain kind of fault. It will serve as an input to our knowledge base defined in the next section. To unclutter notation we will omit $\text{EnvSpec}(x)$ and directly write $f_l(x)$ for the rest of the paper.

Notice, that even though we describe a specific function for bearing fault diagnostics on vibration data, any function f can be used to create the axiom in Eq. (7). Therefore, arbitrary knowledge can be induced and the axiom works for different classification settings.

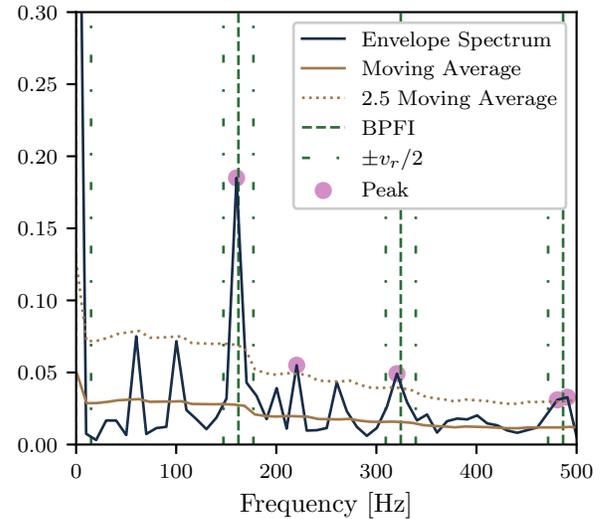


Figure 1. Visualization of the components of the f_{IR} score function for an exemplary signal with an inner race fault (IR). The fault scores of this signal are $f_{\text{OR}} = 0.30$, $f_{\text{IR}} = 0.83$ and $f_{\text{B}} = 0.16$. It would therefore be classified as IR by axiom Eq. (7).

3.2. Identifying Normal Bearings

Healthy bearings can be identified more easily. In comparison to the impulsive signal of a faulty bearing the signal of a normal bearing is rather smooth, see Fig. 2. This difference can be measured by the kurtosis, which is defined as the

fourth standardized moment

$$\text{Kurt}(x) = \frac{1}{n} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s} \right)^4, \quad (8)$$

with mean \bar{x} and standard deviation s of the signal x (Randall & Antoni, 2011). We use this knowledge as a further input to our knowledge base and write the corresponding formula as

$$\forall x ((\text{Kurt}(x) > \text{thr}_{\text{kurt}}) \rightarrow P(x, c_N)), \quad (9)$$

where we set the threshold $\text{thr}_{\text{kurt}} = 0.1$.

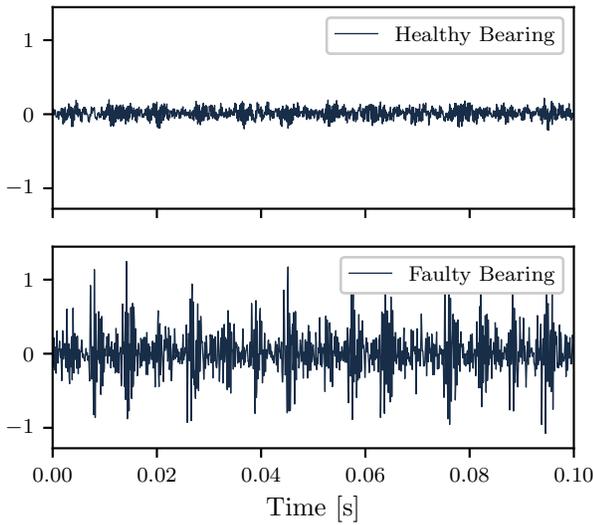


Figure 2. Vibration signal of a healthy and a faulty bearing with a kurtosis of -0.19 and 2.20 respectively.

3.3. Knowledge Base for the LTN

Using these means of diagnosing bearing signals we create a knowledge base as per the definitions for LTNs in Sec. 2.

Domains \mathcal{D} :

data, labels

Variables \mathcal{X} :

$x, x_N, x_{OR}, x_{IR}, x_B$

$\mathbf{D}(x) = \mathbf{D}(x_N) = \mathbf{D}(x_{OR}) = \mathbf{D}(x_{IR}) = \mathbf{D}(x_B) = \text{data}$

Constants \mathcal{C} :

c_N, c_{OR}, c_{IR}, c_B

$\mathbf{D}(c_N) = \mathbf{D}(c_{OR}) = \mathbf{D}(c_{IR}) = \mathbf{D}(c_B) = \text{labels}$

Predicates \mathcal{P} :

$P(x, c)$

$\mathbf{D}_{\text{in}}(P) = \text{data, labels}$

Axioms \mathcal{K} :

Data axioms \mathcal{K}_D :

$\forall x_N P(x_N, c_N)$

$\forall x_{OR} P(x_{OR}, c_{OR})$

$\forall x_{IR} P(x_{IR}, c_{IR})$

$\forall x_B P(x_B, c_B)$

Knowledge axioms \mathcal{K}_K :

$\forall x ((\text{Kurt}(x) > \text{thr}_{\text{kurt}}) \rightarrow P(x, c_N))$

$\forall x ((f_{OR}(x) > \text{thr}_{\text{score}}) \rightarrow P(x, c_{OR}))$

$\forall x ((f_{IR}(x) > \text{thr}_{\text{score}}) \rightarrow P(x, c_{IR}))$

$\forall x ((f_B(x) > \text{thr}_{\text{score}}) \rightarrow P(x, c_B))$

(10)

Groundings \mathcal{G} :

$\mathcal{G}(\text{data}) = \mathbb{R}^n, \mathcal{G}(\text{labels}) = \mathbb{N}^4$

$\mathcal{G}(x_N) \in \mathbb{R}^{m_N \times n}, \mathcal{G}(x_{OR}) \in \mathbb{R}^{m_{OR} \times n}$

$\mathcal{G}(x_{IR}) \in \mathbb{R}^{m_{IR} \times n}, \mathcal{G}(x_B) \in \mathbb{R}^{m_B \times n}$

$\mathcal{G}(x) \in \mathbb{R}^{(m_N + m_{OR} + m_{IR} + m_B) \times n}$

$\mathcal{G}(c_N) = [1, 0, 0, 0], \mathcal{G}(c_{OR}) = [0, 1, 0, 0]$

$\mathcal{G}(c_{IR}) = [0, 0, 1, 0], \mathcal{G}(c_B) = [0, 0, 0, 1]$

$\mathcal{G}(P|\theta) : x, c \mapsto c^T \cdot \text{softmax}(\text{CLF}_\theta(x))$

$\text{thr}_{\text{score}} = 0.49, \text{thr}_{\text{kurt}} = 0.1$

$\text{Kurt}(\mathcal{G}(x)) \in \mathbb{R}$

$\forall l \in \{\text{OR}, \text{IR}, \text{B}\}: f_l(\mathcal{G}(x)) \in [0, 1]$

The subscripts of the variables in \mathcal{X} indicate, that only the fraction, where data has the same label as the subscript, is used. If there is no subscript, then all data is used. This also explains grounding of the different variables onto different dimensional real spaces.

Notice the grounding of predicate P in some classifier CLF, which in turn relies on the parameter configuration θ . The softmax function ensures that a truth value between 0 and 1 is returned by this grounding. As mentioned in Sec. 2 this classifier can be any kind of NN and the parameters of the NN are optimized so that the satisfiability of the axioms \mathcal{K} is maximized.

The knowledge base is specific to our case study of bearing fault diagnosis. But, the general formulation of the whole knowledge base and specifically the knowledge axioms \mathcal{K}_K make it possible to apply this approach to any other problem setting in the fault diagnosis domain.

3.4. Weights for Axiom Groups

Of course, our end goal is not to satisfy the knowledge base developed in Sec. 3.3, but to achieve the highest accuracy on some test set. To improve the performance of the underlying classifier CLF we optimize the satisfiability of the knowledge base, which consists of two kinds of axioms, namely data axioms \mathcal{K}_D and knowledge axioms \mathcal{K}_K . Data axioms rely only on the provided labeled data and are always true in terms of classification accuracy. In contrast the knowledge axioms consist of the heuristics formulated in Sec. 3.1 and 3.2, which are not always true but are based on physical certainties.

If we contemplate on how we as humans learn new things, it mainly starts off with somebody explaining to us how something should work in theory. After understanding this, we go out into the world to apply this knowledge and quickly realize that the theory is based on assumptions, which don't always hold in reality. Therefore we update our beliefs based on new experiences and observations. According to (Fitts & Posner, 1967) this is reflected in the learning phases “cognitive”, “associative” and “autonomous”. We want to apply this way of learning to LTNs.

By weighting the importance of the satisfiability for certain axiom groups differently during the course of training, we can mimic exactly this process. To incorporate this idea we introduce the idea of weighted axiom groups. For this we need to rewrite the satisfiability optimization problem from Eq. (2) to

$$\theta^* = \operatorname{argmax}_{\theta \in \Theta} \sum_{k=1}^K w_k^{(e)} \operatorname{SatAgg}_{\phi \in \mathcal{K}_k} \mathcal{G}(\phi|\theta), \quad (11)$$

with K weights $w_k^{(e)}$ and $\sum_{k=1}^K w_k^{(e)} = 1$, which are dependent on the current training epoch e and a set of K axiom groups \mathcal{K}_k . For our purposes we group axioms into knowledge and data axioms, but other more granular groupings are also possible. We call a weight schedule the setting of weights w_k per axiom group \mathcal{K}_k as a function of epochs e and propose the two weight schedules first knowledge then data (FKTD) and up and down (UAD). The corresponding weight schedules are given in Table 1.

Table 1. Definition of the weight schedules UAD and FKTD for data axioms (D) and knowledge axioms (K) along the training process of N epochs.

Epochs (e)	UAD	FKTD
$e \leq \frac{1}{4}N$	$w_D = 0, w_K = 1$	$w_D = 0, w_K = 1$
$\frac{1}{4}N < e \leq \frac{1}{2}N$	$w_D = \frac{1}{4}, w_K = \frac{3}{4}$	$w_D = 0, w_K = 1$
$\frac{1}{2}N < e \leq \frac{3}{4}N$	$w_D = \frac{3}{4}, w_K = \frac{1}{4}$	$w_D = 1, w_K = 0$
$e > \frac{3}{4}N$	$w_D = 1, w_K = 0$	$w_D = 1, w_K = 0$

3.5. Extending the Feature Space

We can also incorporate our knowledge without using the Real Logic machinery of LTNs by extending the feature space with inputs describing this knowledge (Chao et al., 2022). Hereby we include the score functions (f_{OR}, f_{IR}, f_B) and the Kurtosis of the signals as additional features in the data and train the network based on this enhanced data set. In the following we will use this approach in combination with LTNs and independently as a further baseline on top of networks without any induced knowledge.

4. EXPERIMENTAL SETUP

4.1. Data

We used two bearing fault data sets with different attributes and constant shaft speed for evaluating the proposed approach.

4.1.1. CWRU Data Set

The Case Western University (CWRU) bearing data set² is a widely used benchmark data set for bearing fault classification (Neupane & Seok, 2020). A thorough analysis of the data is given by (Smith & Randall, 2015). The data set consists of vibration signal measurements of the four different classes healthy/normal (N), inner race fault (IR), outer race fault (OR) and rolling (ball) element fault (B). The faults have the different sizes 0.07 in, 0.14 in, 0.21 in or 0.28 in and are measured at different positions. The machine was run on constant shaft speeds of 1730, 1750, 1772 and 1792 rotations per minute (RPM) for different motor loads. For our experiments we will use the drive end data sampled at a frequency of 12 kHz and the baseline data with a frequency of 48 kHz. For the drive end data a SKF 6205-2RS JEM deep groove ball bearing was used, which has the ball pass frequencies BPF_I = 5.415, BPF_O = 3.585 and BSF = 2.357, that we use to create the score from Sec. 3.1. We will exclude the 0.28 in fault data, because a different bearing is used for which neither ball pass frequencies nor measurements for calculating these are available.

4.1.2. MFPT Data Set

As a second data set we used the data provided by the Society for Machinery Failure Prevention Technology³ (MFPT). MFPT provides us with data for healthy bearings and outer race fault conditions with 98 kHz sample frequency and a constant shaft speed of 1560 RPM. Additionally, the data includes inner and outer race fault conditions with 48 kHz sample frequency with the same shaft speed. The test rig used for data generation is equipped with a NICE bearing with the ball pass frequencies BPF_I = 4.755 and BPF_O = 3.245.

²<https://engineering.case.edu/bearingdatacenter>, accessed 02/04/2022

³<https://www.mfpt.org/fault-data-sets/>, accessed: 02/04/2022

4.2. Data Preprocessing

For the CWRU data set we excluded all faults with size 0.28 in because no information about the used bearing is available. From the signal data we extracted time series of length 1200. For the faults, which were sampled at a frequency of 12 kHz, this amounts to an observation time of 0.1 seconds. The baseline data of non faulty bearings was sampled at 48 kHz, therefore we downsampled the data to 12 kHz, so that we have the same time interpretation.

The MFPT data is sampled at different frequencies. The smallest fault frequency is 48 kHz, therefore we downsampled all other signals to this sample rate. Afterwards we extracted time series of length 4800 to also have an observation time of 0.1 seconds.

After the extraction of the time series the scores for different classes and the kurtosis were calculated based on these segments. Since the MFPT data set does not include any ball faults, we excluded the score calculation and the respective axiom from the knowledge base. Depending on whether or not we used feature space extension, we included or excluded the scores and kurtosis from the data used for training and testing of the networks.

4.3. Model Configurations

We used two different kinds of NNs as inputs to the LTN and as baselines for comparison: A Multi Layer Perceptron (MLP) with three fully connected hidden layers, which consist of 32, 32 and 16 nodes, and a CNN consisting of one convolutional layer with kernel size 9 and a fully connected hidden layer with 64 nodes. Both use the exponential linear unit (elu) activation function (Clevert, Unterthiner, & Hochreiter, 2016) with $\alpha = 1$, the Adam optimizer (Kingma & Ba, 2015) with a learning rate of 0.001 and are trained with batches of 32 for 100 epochs. The baseline NNs use the categorical cross entropy loss function. Combining these two NNs with the different means of inducing knowledge and weight schedules, we obtain 16 different models listed in Table 2.

4.4. Experiments

All experiments were run and averaged over the same 7 random seeds. Each model was trained on 10 %, 30 %, 50 %, 70 % and 90 % of the data and test accuracies were calculated on an unseen 10 % of the data. The underlying NNs of the LTNs and for the pure DL approaches were completely identical and were initialized with the same random seed. Thereby both methods suffered or profited from equal initial weight settings likewise. All code was implemented in Python 3 using the Tensorflow and Logic Tensor Networks packages.

5. RESULTS

The results of all experiments for different model configurations for the CWRU data set and the MFPT data set are given in Table 3 and Table 4 respectively. We continue by evaluating the knowledge and the proposed models independently. For each section we discuss both the performance on the CWRU data set and on the one provided by MFPT.

5.1. Pure Knowledge Based Classification

First, we evaluate how the proposed knowledge axioms perform independent of DL on both data sets. We do this by dividing the number of correctly identified labels of a class through the number of all labels of a class. The labels were identified by using the knowledge axioms \mathcal{K}_K (10) from the proposed knowledge base. The results are shown in Table 5. We see that the proposed knowledge axioms work very well on MFPT, with an average value of 0.9 and less so on the CWRU data with an average value of 0.62. From these values we can follow that the MFPT data is very well behaved in terms of characteristic signals for different fault types and that these are not significantly masked by other influences. For the CWRU data on the other hand, there appear to be multiple other influences on the signals, which corresponds to one of the key findings by (Smith & Randall, 2015). Additionally, we see that faults of type B are very hard to classify with only the proposed scoring function.

For MFPT the purely knowledge based approach even outperformed all analyzed neural networks regardless of the amount of data used, which further underlines the validity of the proposed logical axioms for clear unmasked bearing fault data. On the other hand, when using the CWRU data set the purely knowledge based classification is outperformed by all analysed models if 30 % or more of the data was used.

5.2. Real Logic in the Loss Function

Next, we evaluate the performance of LTNs (LtnMlpNoF & LtnCnnNoF) given the proposed knowledge base from Sec. 3.3 against the pure DL approaches with a MLP or CNN (NnMlpF & NnCnnF).

On the CWRU data set LTNs outperformed their respective NN counterparts along all data fractions. The difference is especially noticeable for the MLP, where the average test accuracy of LTNs was increased by 0.02.

When using the MFPT data set no clear difference in performance can be read from the experimental data. Depending on the data fraction either LTNs or NNs performed better.

5.3. Extending the Feature Space

Now, we extend the feature spaces for both LTNs (LtnMlpNoT & LtnCnnNoT) and NNs (NnMlpT & NnCnnT). With

Table 2. Names and configurations of the models used for experiments.

Name	Model	Neural Network	Weight Schedule	Extend Features Space
LtnMlpNoT	LTN	MLP	None (NO)	True
LtnMlpNoF	LTN	MLP	None (NO)	False
LtnMlpUadT	LTN	MLP	Up And Down (UAD)	True
LtnMlpUadF	LTN	MLP	Up And Down (UAD)	False
LtnMlpFktdT	LTN	MLP	First Knowledge Then Data (FKTD)	True
LtnMlpFktdF	LTN	MLP	First Knowledge Then Data (FKTD)	False
LtnCnnNoT	LTN	CNN	None (NO)	True
LtnCnnNoF	LTN	CNN	None (NO)	False
LtnCnnUadT	LTN	CNN	Up And Down (UAD)	True
LtnCnnUadF	LTN	CNN	Up And Down (UAD)	False
LtnCnnFktdT	LTN	CNN	First Knowledge Then Data (FKTD)	True
LtnCnnFktdF	LTN	CNN	First Knowledge Then Data (FKTD)	False
NnMlpT	NN	MLP	-	True
NnMlpF	NN	MLP	-	False
NnCnnT	NN	CNN	-	True
NnCnnF	NN	CNN	-	False

Table 3. Test accuracies for different models and data fractions used for training on the CWRU data set. The standard deviation is given in parenthesis. Bold entries mark the best performing model for the respective data fraction.

Model	10 % of Data	30 % of Data	50 % of Data	70 % of Data	90 % of Data
LtnMlpNoT	0.595 (0.032)	0.731 (0.016)	0.8 (0.021)	0.84 (0.014)	0.874 (0.015)
LtnMlpNoF	0.526 (0.026)	0.678 (0.015)	0.74 (0.015)	0.782 (0.012)	0.799 (0.016)
LtnMlpUadT	0.598 (0.028)	0.748 (0.021)	0.81 (0.014)	0.859 (0.019)	0.891 (0.013)
LtnMlpUadF	0.533 (0.042)	0.675 (0.018)	0.734 (0.019)	0.777 (0.008)	0.807 (0.016)
LtnMlpFktdT	0.597 (0.035)	0.74 (0.016)	0.799 (0.019)	0.845 (0.016)	0.868 (0.02)
LtnMlpFktdF	0.552 (0.022)	0.668 (0.017)	0.727 (0.019)	0.773 (0.016)	0.8 (0.012)
LtnCnnNoT	0.581 (0.018)	0.778 (0.011)	0.855 (0.014)	0.896 (0.011)	0.925 (0.012)
LtnCnnNoF	0.545 (0.027)	0.727 (0.022)	0.818 (0.011)	0.858 (0.017)	0.89 (0.019)
LtnCnnUadT	0.635 (0.027)	0.82 (0.012)	0.89 (0.012)	0.917 (0.01)	0.934 (0.011)
LtnCnnUadF	0.584 (0.029)	0.747 (0.017)	0.814 (0.017)	0.863 (0.009)	0.876 (0.015)
LtnCnnFktdT	0.641 (0.024)	0.826 (0.016)	0.886 (0.018)	0.904 (0.018)	0.915 (0.023)
LtnCnnFktdF	0.591 (0.023)	0.759 (0.02)	0.824 (0.01)	0.856 (0.015)	0.882 (0.012)
NnMlpT	0.551 (0.025)	0.69 (0.027)	0.762 (0.011)	0.807 (0.016)	0.847 (0.009)
NnMlpF	0.496 (0.031)	0.649 (0.028)	0.721 (0.022)	0.762 (0.01)	0.787 (0.011)
NnCnnT	0.535 (0.02)	0.738 (0.02)	0.851 (0.017)	0.902 (0.013)	0.933 (0.012)
NnCnnF	0.477 (0.036)	0.72 (0.021)	0.802 (0.017)	0.854 (0.008)	0.884 (0.011)

feature space extension, we see a similar picture as in Sec. 5.2, but the differences are more pronounced.

For the CWRU data set LtnMlpNoT outperformed its counterpart NnMlpT by between 0.03 and 0.04 points, where out-performance is especially high for smaller data fractions and lower when using more data. When examining LtnCnnNoT and NnCnnT we see a similar dynamic. The LTN has a higher accuracy for data fractions 0.1 and 0.3 but the gap is closed for larger data fractions, where the NN even outperformed the LTN slightly.

Again, MFPT data does not show as much of a difference between the models. We see, that LtnMlpNoT outperformed NnMlpNoT marginally along all data fractions. For the CNN based model on the other hand NnCnnT performed better on less training data. Starting with a data fraction of 0.5 the LtnCnnNoT increased its accuracy in comparison to the NN and

had a 0.04 higher accuracy when all data was used.

In general, just by extending the feature space the performance of all models increased significantly. For the CWRU data the inclusion of our proposed knowledge features into the benchmark NN increased test accuracy by 0.04 on average and for the MFPT data by 0.02.

5.4. Weight Schedules for Axioms Groups

Finally, we compare different weight schedules for data and knowledge axioms in LTNs (LtnMlpUadT, LtnMlpFktdT & LtnCnnUadT, LtnCnnFktdT).

By using the weight schedule UAD the performance of LtnMlpNoT was further improved by an average of 0.01 on the CWRU data set. FKTD on the other hand did not seem to increase performance. The same can be said when the fea-

Table 4. Test accuracies for different models and data fractions used for training on the MFPT data set. The standard deviation is given in parenthesis. Bold entries mark the best performing model for the respective data fraction.

Model	10 % of Data	30 % of Data	50 % of Data	70 % of Data	90 % of Data
LtnMlpNoT	0.547 (0.032)	0.63 (0.031)	0.682 (0.029)	0.732 (0.021)	0.772 (0.036)
LtnMlpNoF	0.5 (0.019)	0.596 (0.036)	0.636 (0.033)	0.698 (0.032)	0.741 (0.022)
LtnMlpUadT	0.548 (0.03)	0.63 (0.03)	0.679 (0.031)	0.719 (0.032)	0.777 (0.027)
LtnMlpUadF	0.521 (0.042)	0.58 (0.028)	0.653 (0.029)	0.698 (0.017)	0.733 (0.029)
LtnMlpFktdT	0.552 (0.045)	0.649 (0.032)	0.688 (0.031)	0.723 (0.023)	0.781 (0.047)
LtnMlpFktdF	0.494 (0.019)	0.583 (0.024)	0.645 (0.03)	0.697 (0.018)	0.74 (0.029)
LtnCnnNoT	0.475 (0.054)	0.613 (0.056)	0.692 (0.02)	0.751 (0.025)	0.802 (0.033)
LtnCnnNoF	0.461 (0.071)	0.551 (0.043)	0.668 (0.034)	0.725 (0.036)	0.778 (0.056)
LtnCnnUadT	0.491 (0.07)	0.537 (0.063)	0.631 (0.029)	0.634 (0.109)	0.64 (0.179)
LtnCnnUadF	0.469 (0.06)	0.531 (0.048)	0.534 (0.056)	0.569 (0.099)	0.601 (0.116)
LtnCnnFktdT	0.479 (0.047)	0.56 (0.039)	0.616 (0.046)	0.648 (0.081)	0.664 (0.139)
LtnCnnFktdF	0.49 (0.028)	0.477 (0.056)	0.519 (0.083)	0.534 (0.054)	0.571 (0.089)
NnMlpT	0.542 (0.041)	0.618 (0.024)	0.677 (0.027)	0.706 (0.028)	0.756 (0.036)
NnMlpF	0.512 (0.03)	0.58 (0.028)	0.643 (0.023)	0.689 (0.034)	0.739 (0.033)
NnCnnT	0.502 (0.059)	0.615 (0.032)	0.684 (0.019)	0.72 (0.027)	0.757 (0.036)
NnCnnF	0.503 (0.055)	0.604 (0.029)	0.657 (0.025)	0.692 (0.021)	0.752 (0.031)

Table 5. Fraction of correctly classified data when adhering to the underlying axioms of the proposed knowledge base per class and data set.

Underlying Axiom	CWRU	MFPT
$\forall x ((Kurt(x) > thr_{Kurt}) \rightarrow P(x, c_N))$	0.91	0.88
$\forall x ((f_{OR}(x) > thr_{score}) \rightarrow P(x, c_{OR}))$	0.49	1.00
$\forall x ((f_{IR}(x) > thr_{score}) \rightarrow P(x, c_{IR}))$	0.76	0.83
$\forall x ((f_B(x) > thr_{score}) \rightarrow P(x, c_B))$	0.32	-

ture space was not extended. Without feature extension we only see an increase in the standard deviation of results, especially for the weight schedule FKTD. For LTNs based on the CNN the addition of weighted axioms shows stronger impact on accuracy. Especially the lower data fractions profited from the weights, but this improvement gradually levels out when using more training data. This is true for both extended and not extended feature spaces. Again, the UAD weight schedule performed best and increased the performance by up to 0.05.

The MFPT data gives us a rather different picture. For MLPs there are close to no performance gains for both extended and not extended features spaces when weighted axioms were used. CNNs even show a drastic deterioration of model performance when weights were included. This observation is paired with a sharp increase of the standard deviation to a value of up to 0.18 over the seven analyzed runs, which shows that the performance was highly dependent on the random seed and data composition. Fig. 3 shows that the LTN seems to have gotten stuck in a local optimum when trained on one of the low performing seeds. A similar training progress was observed for all other low performing seeds. It seems, that the concentration on the knowledge axioms at the beginning of the training found a local optimum, which is hard to es-

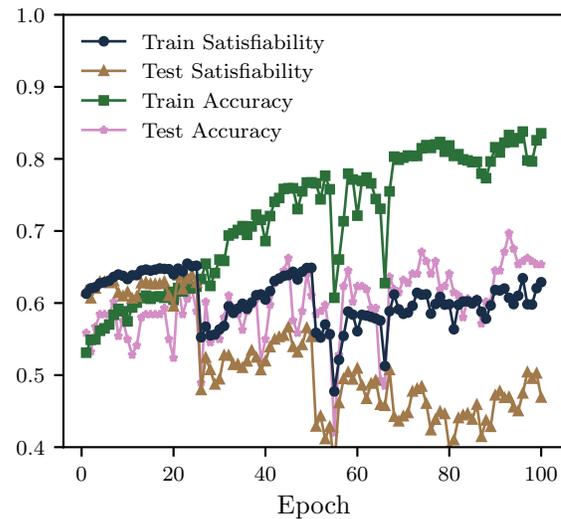


Figure 3. Training of LtnCnnUadT for 100 epochs on 90 % of the MFPT data set for a bad performing random seed. The train satisfiability seems to be stuck in a local optimum.

cape.

5.5. Discussion

Although the improvement of each single aspect (real logic in the loss function, extending the feature space and weight schedules) of LTNs in comparison to the pure DL approaches does not seem large, the combination of these show a significant increase in accuracy for LTNs. We identify the optimal models LtnMlpUadT and LtnCnnUadT on the CWRU data set with an increase in accuracy of up to 0.10 for small amounts of training data in comparison to the feature ex-

tended NNs. On the MFPT data set the best performing models where LtnMlpNoT and LtnCnnNoT with a maximum performance increase of 0.04. Note that compared to the networks without the use of our proposed knowledge features the performance increased by up to 0.15.

Especially on the CWRU data we see, that the gain decreases with the amount of data used. This aligns with one of the main motivations for hybrid models, namely the reduction of necessary labeled data and further underlines the strength of purely data driven DL, when enough data is available. Weight schedules, which we motivated by the way humans learn, also increased the test accuracy. Notably, the strongest improvement could be observed when focusing on knowledge first and putting more and more weight on the contribution given by the data.

We also observe a surprising difference between the performance increase of the analyzed data sets. Even though the induced knowledge was better in terms of per class accuracy for the MFPT then for the CWRU data set, the gains in test accuracy were mainly seen for the CWRU data. An explanation for this can maybe be found in the very bad performance of weighted schedules on the MFPT data set. It seems like the focus on the knowledge brought the LTN to a local optimum, which is hard to get out of. Maybe the nearly perfect knowledge limits the search space for optimal parameter configurations to a degree, that it can't be optimized through more data. Or the underlying models where not appropriate for the data set and the weight schedules.

6. CONCLUSION

We proposed an application of LTNs for fault diagnostics in the special case of bearing faults with constant shaft speed. To this end, we introduced a scoring function based on physical attributes of the bearings as a representation of expert knowledge and extended the LTN framework by weight schedules. Because of the general formalization of the LTN knowledge base the method can readily be applied to various other diagnostic tasks with domain-specific adjustments of the scoring function. Our proposed approach was evaluated on two different data sets and showed an increase in test accuracy in comparison to the benchmark NNs, which were, in some experiments, also enhanced with the created knowledge features. We demonstrated that inducing knowledge increased performance, particularly when less data was available and can also be used to gain a better understanding of the data set, e.g., getting an indication on whether fault signals are masked by other influences. However, we also showed that the performance gains of LTNs are not completely intuitive and that LTNs with weight schedules can get stuck in local minima during training for certain data compositions.

The findings from this work confirm the hypothesis that a combination of DL with expert knowledge in the domain of

fault diagnosis is a promising direction, and motivate further studies and future research. Most notably, the proposed weight schedules need to be analyzed in more detail, since they have a positive impact on the training performance in most situations, but cause a stagnation of the learning curve in other situations. In this context, the interplay between different NN architectures, weight schedules, axioms and data sets needs to be studied further to leverage the full potential of LTNs. Also a factor that could further strengthen the performance is the utilization of more complex logical formulas and logical reasoning in the loss function, which is a feature of logic-based knowledge representation but has not yet been exploited in this research. Currently the scoring function represents physical knowledge from the domain of bearing fault diagnosis, but it can be reconfigured to incarnate physical knowledge from other domains as well. A more abstract representation with clear pathways to instantiate the scoring function in different domains would also be part of future research.

REFERENCES

- Arinez, J. F., Chang, Q., Gao, R. X., Xu, C., & Zhang, J. (2020). Artificial intelligence in advanced manufacturing: Current status and future outlook. *Journal of Manufacturing Science and Engineering*, 142(11).
- Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... others (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58, 82–115.
- Badreddine, S., Garcez, A. d., Serafini, L., & Spranger, M. (2022). Logic tensor networks. *Artificial Intelligence*, 303.
- Bianchi, F., & Hitzler, P. (2019). On the capabilities of logic tensor networks for deductive reasoning. In *Aai spring symposium: combining machine learning with knowledge engineering*.
- Bonnardot, F., Randall, R., Antoni, J., & Guillet, F. (2004). Enhanced unsupervised noise cancellation using angular resampling for planetary bearing fault diagnosis. *International journal of acoustics and vibration*, 9(2), 51–60.
- Chao, M. A., Kulkarni, C., Goebel, K., & Fink, O. (2022). Fusing physics-based and deep learning models for prognostics. *Reliability Engineering & System Safety*, 217.
- Clevert, D., Unterthiner, T., & Hochreiter, S. (2016). Fast and accurate deep network learning by exponential linear units (elus). In *4th international conference on learning representations, conference track proceedings*.
- Donadello, I., Serafini, L., & Garcez, A. d. (2017). Logic tensor networks for semantic image interpretation. In *Proceedings of the twenty-sixth international joint con-*

ference on artificial intelligence, *IJCAI-17* (pp. 1596–1602).

- Fink, O., Wang, Q., Svensen, M., Dersin, P., Lee, W.-J., & Ducoffe, M. (2020). Potential, challenges and future directions for deep learning in prognostics and health management applications. *Engineering Applications of Artificial Intelligence*, 92.
- Fitts, P. M., & Posner, M. I. (1967). *Human performance*. Brooks/Cole Publishing Company.
- Garcez, A. d., & Lamb, L. C. (2020). Neurosymbolic ai: the 3rd wave. *arXiv preprint arXiv:2012.05876*.
- Hájek, P. (2013). *Metamathematics of fuzzy logic* (Vol. 4). Springer Science & Business Media.
- Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd international conference on learning representations, conference track proceedings*.
- Liu, R., Yang, B., Zio, E., & Chen, X. (2018). Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*, 108, 33–47.
- Marcus, G. (2020). The next decade in ai: four steps towards robust artificial intelligence. *arXiv preprint arXiv:2002.06177*.
- Neupane, D., & Seok, J. (2020). Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: A review. *IEEE Access*, 8.
- Randall, R. B., & Antoni, J. (2011). Rolling element bearing diagnostics—a tutorial. *Mechanical systems and signal processing*, 25(2), 485–520.
- Selcuk, S. (2017). Predictive maintenance, its implementation and latest trends. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 231(9), 1670–1679.
- Smith, W. A., & Randall, R. B. (2015). Rolling element bearing diagnostics using the case western reserve university data: A benchmark study. *Mechanical systems and signal processing*, 64, 100–131.
- Steenwinckel, B., De Paepe, D., Hautte, S. V., Heyvaert, P., Bentefrit, M., Moens, P., ... others (2021). Flags: A methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing expert

knowledge with machine learning. *Future Generation Computer Systems*, 116, 30–48.

- Wang, Q., Taal, C., & Fink, O. (2021, 07). Integrating expert knowledge with domain adaptation for unsupervised fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*.
- Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115, 213–237.

BIOGRAPHIES



Maximilian-Peter Radtke studied business mathematics at the University of Mannheim, Germany and graduated in 2018. After his studies he worked as a data science consultant in various industries for two and half years before returning to academia. Since 2021 he has been working at the Technische Hochschule Ingolstadt (THI) as part of AIMotion Bavaria and the research group AI applications for innovative production and logistic systems. His research interests include the combination of symbolic and sub-symbolic AI approaches and the incorporation of knowledge into deep learning in the area of fault diagnostics and prognostics.



Jürgen Bock is a computer scientist, who graduated as Diplom-Informatiker from Ulm University, Germany, and as Bachelor of Information Technology with Honours from Griffith University, Brisbane, Australia, in 2006. He began his research career at the FZI Research Center for Information Technology in Karlsruhe, Germany, and received his PhD from the Karlsruhe Institut of Technology (KIT) in 2012. After 2 years as post doc and team leader at the FZI, he joined the corporate research department of KUKA Robotics in Augsburg, Germany as developer and later leader of the team Smart Data and Infrastructure. In 2020 he joined the Technische Hochschule Ingolstadt (THI) as research professor in the area of AI applications in innovative production and logistics systems.

Domain Adaptation in Predicting Turbocharger Failures Using Vehicle's Sensor Measurements

Mahmoud Rahat¹, Peyman Sheikholharam Mashhadi¹, Sławomir Nowaczyk¹, Thorsteinn Rögnvaldsson¹, Atabak Taheri², and Ataollah Abbasi²

¹ *Center for Applied Intelligent Systems Research (CAISR) Halmstad University, Sweden*
{mahmoud.rahat, peyman.mashhadi, slawomir.nowaczyk, thorsteinn.rognvaldsson}@hh.se

² *Volvo Group Trucks Technology Gothenburg, Vastra Gotaland County, Sweden*
{atabak.taheri, ataollah.abbasi}@volvo.com

ABSTRACT

The discrepancy in the distribution of source and target domains is usually referred to as a domain shift. It is one of the reasons for the inferior performance of machine learning solutions at deployment. We illustrate that the domain shift issue is pertinent to the readings of the vehicles' operational sensors. This is due to the fact that these measurements are collected over a period of time and are susceptible to various changes that happen in the meantime. Examples of these changes are usage pattern variations, aging of the vehicles, seasonal shifts, and driver changes. However, domain adversarial neural networks (DANN) have shown promising results to reduce the negative impact of the domain shift. The present study investigates domain adaptation (DA) in the predictive maintenance field by estimating the remaining useful life (RUL) of turbochargers. The devices are operating on a fleet of VOLVO trucks, and the information about their services is collected over four years between 2016 and 2019. The input features to the model are a set of bi-weekly collected measurements called logged vehicle data (LVD). The contributions of this paper are two-fold. First, we propose a new approach for detecting domain (covariate) shift using an autoencoder. Second, we adapt domain adversarial neural networks to the specific application of predicting turbocharger failures. Finally, we deploy a recurrent feature extraction layer in the DANN architecture to incorporate temporal aspect of the data. The experimental results demonstrate the superiority of the proposed method over the traditional approach.

Mahmoud Rahat et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

The main goal of predictive maintenance (PdM) is to monitor equipment behavior and suggest a proper time for performing maintenance. This provides multiple benefits for businesses by avoiding unexpected breakdowns and improving the quality of customer service. When it comes to heavy-duty vehicles such as trucks, PdM's importance becomes more evident as they usually carry out important tasks around the clock. Any potential breakdown could lead to catastrophic situations in terms of cost, time, and pledges.

Numerous papers in the literature use data-driven approaches and machine learning for predictive maintenance of industrial equipment (W. Zhang, Yang, & Wang, 2019). Based on how they formulate the problem, these methods could coarsely be divided into two groups of regression (Ding, Jia, Miao, & Huang, 2021; Y. Zhang, Hutchinson, Lieven, & Nunez-Yanez, 2020) and classification approaches (Prytz, Nowaczyk, Rögnvaldsson, & Byttner, 2015; Rahat, Pashami, Nowaczyk, & Kharazian, 2020). In the regression approach, the methods estimate a component's RUL as a value of a continuous quantity (unbounded real number). In this scenario, the RUL estimates the amount of time a piece of equipment can perform its intended functionality. However, one could argue that estimation of the exact remaining useful life is too complicated and unnecessary as the only important question that we need to answer each time a truck visits a workshop is binary, either to substitute the component or not (Prytz et al., 2015; Rahat et al., 2020).

By looking at the literature, one can observe that methods based on neural networks and ensemble frameworks are becoming more popular in the field (Mashhadi, Nowaczyk, & Pashami, 2020; Revanur, Ayibiowu, Rahat, & Khoshkangini, 2020; Xia, Song, Zheng, Pan, & Xi, 2020; Rahat et al., 2020; Uddagiri, Ramalingam, Rahat, & Mashhadi, 2021) and generally, the predictive maintenance systems are becoming more

sophisticated over time. Although most of the initial works on PdM were focused on basic components such as industrial bearings (Wang, Liang, Zheng, Gao, & Zhang, 2020), it is clear that the trend is going toward applying PdM on more complex equipment such as batteries (Altarabichi, Fan, Pashami, Mashhadi, & Nowaczyk, 2021), compressors (Fan, Nowaczyk, & Rögnavaldsson, 2015a, 2015b), and turbochargers (Rahat et al., 2020).

The underlying assumption of the machine learning models is that the train and test data are sampled independently from a static distribution that does not change between learning and evaluation. This is generally considered as a necessary condition that ensures the likelihoods the model receives match the expectations within the same distribution. However, this assumption usually does not hold for real-world time-series since such a data is collected under different working conditions overtime. The key motivation of the paper is to formulate changes as such in the context of domain shift and obtain versatility using power of DANN.

Because of the natural temporal order in time-series, it is not possible to shuffle the data randomly between train and test, since we must avoid learning from future observations and evaluating on the past. This has also been considered as part of the sample selection bias (Quiñonero-Candela, Sugiyama, Schwaighofer, & Lawrence, 2008). The significance of applying domain adaptation in the context of time-series data becomes more clear once we consider further challenges such as seasonality, and usage pattern shifts. Knowledge transfer between different domains has recently gained a lot of attention in several neural network applications such as natural language processing (Yang et al., 2019), and machine vision (Kharazian, Rahat, Fatemizadeh, & Nasrabadi, 2020). The failure prognosis and fault diagnosis applications are not an exception to the mentioned trend (Che, Deng, Lin, Hu, & Hu, 2021; Li, Tang, Tang, & He, 2021; Mao, Liu, Ding, Safian, & Liang, 2020).

The two central contributions of the paper are 1) proposing a new approach for domain shift detection using an autoencoder 2) adapting DANN to the specific application of predicting turbocharger failures. Moreover, we use a recurrent feature extraction layer to integrate sequential information in the common feature extraction layers. Previous works have shown the effectiveness of using an LSTM (as feature extraction) in the context of DANN for estimating the remaining useful life (da Costa, Akçay, Zhang, & Kaymak, 2020).

We first analyze the existence of domain shift in the data by using an autoencoder. Then, we address the issue by applying unsupervised domain adaptation. The adopted network architecture simultaneously optimizes two heads; a regressor that estimates the RUL (the primary task) and a binary classifier that predicts whether the samples are drawn from source or target distribution (the auxiliary task). This architecture

promotes the emergence, in the shared internal representation layer, of features that remain robust over time. In other words, it helps to avoid overfitting to the source-specific features. We further improve the results by considering a sequence of observations using an LSTM network in the feature extraction layers. The experiments demonstrate that the RUL prediction error reduces significantly over different data splits once we add domain adaptation head to the model.

The remainder of the paper is organized as follows. Section 2 introduces the data. Section 3 presents the proposed methodology. Section 4 demonstrates the results, and finally section 5 concludes the paper.

2. DATA

The data used in this study includes bi-weekly measurement readouts of the sensors installed on 415 VOLVO trucks. These measurements are called logged vehicle data or, in short, LVD and are collected over four years between 2016 and 2019. Most of the measurements are collected through telecommunication and during workshop visits. The original data had unequal timestamps and contained missing values. The missing values are imputed using mean, and linear interpolation is used to equalize the reading intervals. The data includes 372 attributes in total.

The date and time of the measurement, average speed, mileage, vehicle identification number (id), time driving in each gear are among the features available in the dataset. The information about the repairs done on the turbochargers of the trucks is stored in a separate table called Vehicle Service Records (VSR). This table includes vehicle id, part code, and repair dates. The vehicle identifier field (id) is common between the repair and LVD tables and can be used to join two tables.

3. METHODOLOGY

In this section we first, in 3.1, introduce the new approach for domain (dataset) shift detection using autoencoder, and then, in 3.2, we present the adapted model for predicting failures of turbocharger in the presence of such domain shift.

3.1. Domain shift detection using autoencoder

One of the most common types of domain (dataset) shifts is called covariate shift where input sample distribution $P(x)$ is subject to changes (Quiñonero-Candela et al., 2008). A traditional approach to detect such a shift is to label source and target domain samples with zero and one respectively and then train a binary classifier on the newly labeled dataset. A higher performance in the trained model indicates a higher shift in the dataset. As an alternative, we propose to use an autoencoder for detecting the shift in the dataset. Autoencoders lend themselves naturally to this purpose since they consider only reconstruction of input features which is exactly the marginal

distribution required in detecting the covariate shift (change in $P(x)$). Another advantage of using autoencoder for detecting covariate shift is that the effect can also be detected based on the discrepancy between the representation of source and target in the bottleneck layer.

An autoencoder consists of two parts, an encoder $f(x)$ that maps the input x into a lower dimensional representation and a decoder $g(x)$ that reproduces back the original input from the learned internal representation. Therefore, the objective of the network is to minimize the loss function formulated in equation (1) w.r.t the parameters of the model (Θ_f and Θ_g).

$$\ell(\Theta_f, \Theta_g) = \sum_{i=1}^N \|x_i - g(f(x_i))\|^2 \quad (1)$$

We propose to consider the Average Reproduction Error (in short ARE) as a quantitative indicator of the magnitude of domain shift in the dataset, see equation (2). As an example, let's say we have two datasets are called source S and target T . We train autoencoder on S and evaluate it on T , by calculating the ARE. The higher the value of ARE, the higher the shift between S and T distributions. To the best of our knowledge, this has not been practiced before for measuring the shift between two distributions.

$$ARE = \frac{\ell(\Theta_f, \Theta_g)}{N} \quad (2)$$

3.2. Predicting turbocharger failures

The prediction model used in this paper is inspired from the domain adaptation technique introduced in (Ganin et al., 2016). Figure 1 shows the adapted structure of the DANN model. The layers on the left side of the figure represent feature extraction layers parameterized in Θ_f . The produced internal representation is then mapped into two separate heads. The upper head performs the main task of RUL prediction (parameterized in Θ_r). Thus, it is essentially a regressor mapping the internal representation to a positive unbounded number. The lower head represents the domain adaptation branch of the network parameterized in Θ_d . This branch is connected to the feature extractor through a gradient reversal layer (Ganin et al., 2016).

The input to the network is a sample with a pair of labels $(x_i, [r_i, d_i])$ where x_i is drawn from $Source \cup Target$ distribution randomly, and r_i and d_i represent the ground truth for the remaining useful life (y_{rul}) and the ground truth domain (y_{domain}) respectively. If x_i is drawn from $Target$ domain (i.e. $d_i = 1$), then the task branch will be turned off and r_i won't be considered accordingly. Equation (3) represents the loss function of the network. The negative sign in the for-

mula is to make sure the optimization algorithm maximizes the domain classification error.

$$\ell(\Theta_f, \Theta_r, \Theta_d) = \frac{1}{N} \sum_{i=1}^{N=N_s+N_d} \ell_r^i(\Theta_f, \Theta_r, d_i) - \alpha \left(\frac{1}{N} \sum_{i=1}^{N=N_s+N_d} \ell_d^i(\Theta_f, \Theta_d) \right), \quad (3)$$

where α is a parameter to gauge the effect of adversarial branch and is tuned adaptively, and $\ell_r^i(\Theta_f, \Theta_r, d_i)$ calculates loss on the task branch (RUL prediction) with respect to the target values (r_i) and predicted values (\hat{r}_i) and is calculated using formula (4).

$$(1 - d_i) \cdot |r_i - \hat{r}_i|^2 \quad (4)$$

and $\ell_d^i(\Theta_f, \Theta_d)$ applies cross entropy loss for a binary classification task and is calculated using formula (5).

$$d_i \cdot \log(\hat{d}_i) + (1 - d_i) \cdot \log(1 - \hat{d}_i) \quad (5)$$

where \hat{d}_i represents the predicted domain of i^{th} sample by the domain branch of the network, d_i indicates the ground truth domain label, and N_s and N_d are the number of samples drawn from source and target distributions. The optimization algorithm in the network simultaneously minimizes the loss on the task branch while maximizing the loss on the domain adaptation branch. The intuition behind maximizing the loss on the domain adaptation branch is to promote emergence of the features that are uninformative in distinguishing between source and target samples. This will ensure the network does not overfit to the source-specific features, and keeps its generalization capacity when transferred to the target domain.

One important aspect of analyzing the time series data is the information available in the sequence of observations. Although the adopted DANN architecture gains boost by introducing the domain adaptation branch, it is still unable to process a sequence of observations as the feature extraction layers only accepts a single readout at a time.

To mitigate this drawback, a recurrent neural network layer known as long short-term memory (LSTM) is used in the feature extraction layer. The input samples to this network are sequences of observations $([x_i^{t-k}, \dots, x_i^t], [r_i, d_i])$ where again $[x_i^{t-k}, \dots, x_i^t]$ is a sequence of observations drawn randomly from $Source \cup Target$ distributions. While the DANN model receives a single readout as input, the recurrent archi-

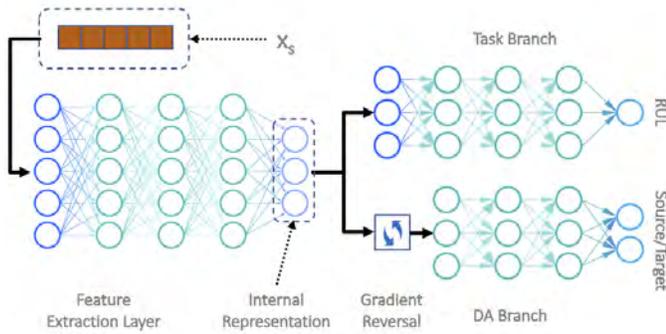


Figure 1. DANN architecture

ecture receives a sequence of observations with length k (in the experiments we consider $k = 8$). Figure 2 shows the structure of the recurrent model. As you can see, the task branch and the domain adaptation branch are similar to the regular DANN.

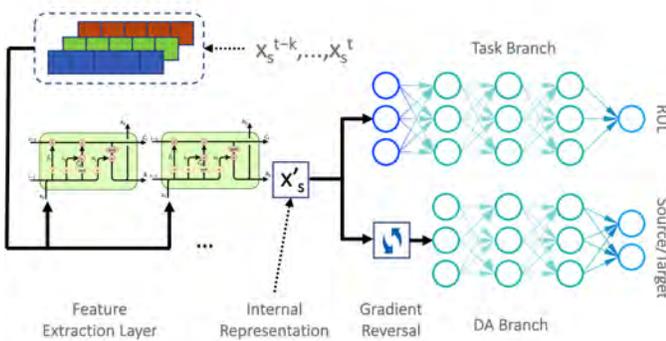


Figure 2. recurrent DANN architecture

4. EXPERIMENTS AND DISCUSSIONS

This section is split into four subsections. 4.1 studies usage of autoencoder for shift detection. 4.2 and 4.3 demonstrate the results achieved by domain adaptation model with single observation and sequential observations. Finally, 4.4 visualizes the effect of domain adaptation in the feature extraction layer by plotting the output of internal representation layer.

4.1. Domain shift in the data

In the first experiment, we conduct an empirical test on the LVD readouts to investigate the existence of distribution change between source and target domains. A three-layer autoencoder neural network is employed in a self-supervised setup to map the sensor readouts into a bottleneck layer (size of 32 neurons) and reproduce the input signals back from the internal representation. Here, we are interested to compare the Average signal reproduction error (see section 3.1) between

source and target domains. The Average reproduction error is used as a quantitative empirical indicator of the magnitude of domain shift between different domains.

In order to study the effect of time on the distribution change of input features $P(x)$, we divide the data into four sections using three split dates each six month apart from each other (see figure 3). The first split represents the source domain and is used to train the autoencoder. The three other sections are used as target domains for evaluating the autoencoder where target1 covers six month after the source time span. Target2 covers the time between six months to one year after the source time span. Eventually, target3 starts one year after the source time span.

We first train the autoencoder using source data and then evaluate it using three target distributions each time calculating the mean absolute error between the regenerated signal values and the original values.



Figure 3. The figure shows how the data is split between source and target domains. The time interval between split dates is six months. The reason for proposing multiple targets is to evaluate the magnitude of the shift in the data compared to source distribution with respect to the time elapse.

Figure 4 shows the learning curves of the autoencoder. The blue curve indicates the training loss on the source domain, while the red, green, and yellow curves illustrate the loss on target1, target2, and target3 respectively. The y axis represents the reconstruction loss, and the x axis shows the training epochs.

The first important observation from the learning curves is that the amount of loss increases comparing target1 to target2 and target3. This illustrates that as we go further away from the source time span, the trained autoencoder quickly losses its generalization capability and is no longer able to accurately regenerate the input signals. Another interesting observation is that the trained autoencoder performs perfectly well on target1 which means the amount of shift is subtle up to six month after the training time. However, the situation changes quickly as we go further away from training period creating a huge contrast comparing the losses of target3 and target1.

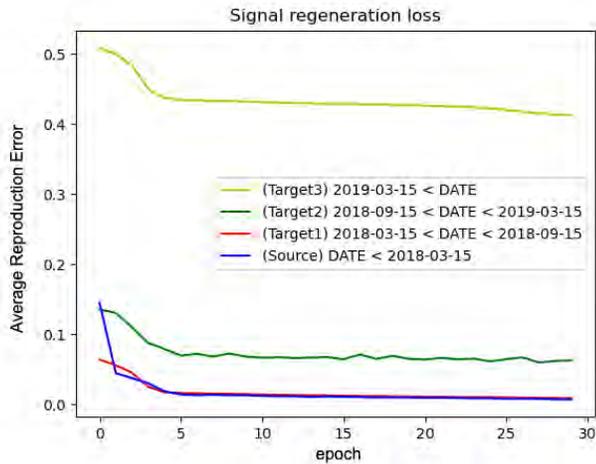


Figure 4. Comparing source and three target losses while generating input signals using an autoencoder. The red curve shows the average loss for a period of six months after source, the green after one year, and the yellow for more than a year. Note how the loss increases over time comparing target1, target2, and target3.

We interpret this variance as an indication of a distributional shift in the data over time. This shift can dwindle the performance of any data-driven model that tries to predict future by learning from the past values of the signals. Therefore, we suggest the application of domain adaptation.

4.2. Domain adaptation without sequential memory

The previous experiment demonstrated the existence of the distribution shift in the data. In this experiment, we investigate the effect of transductive learning through domain adaptation of source and target features. We do not include the sequence information in the first setup and consider each readout as a stand-alone observation. The effect of adding memory to the model is analyzed afterwards in section 4.3.

A dense feed-forward neural network architecture with two output heads is employed to model the remaining useful life of turbochargers. The network has an input layer with 372 neurons, followed by two feature extraction layers with 100 and 64 neurons. The extracted features are then mapped into two separate heads, the RUL regressor and the domain predictor. The RUL regressor has two layers with corresponding 50 and 1 neurons. In comparison, the domain predictor contains four layers starting with a gradient reversal layer followed by 64, 32, and 2 neurons in the subsequent three layers. All layers use the ReLu activation function except the last layer of the regressor, which uses linear activation function and the last layer of the domain predictor uses softmax binary cross entropy. The Adam optimizer with learning rate equal to 0.001 is used during the training phase. The batch size is 128. Finally, the number of training epochs is tuned

Table 1. The evaluation results comparing the RUL predictions with or without the domain adaptation respectively presented in "NN" and "DANN" columns. The values show average and standard deviation of five times running the experiment with different random seeds.

Split	Split date	NN (MAE)	DANN (MAE)
1	2017-12-15	159.68±2.59	122.50±2.25
2	2018-01-15	148.71±4.49	116.73±2.50
3	2018-02-15	155.15±4.80	119.46±3.33
4	2018-03-15	151.69±6.44	116.74±2.08

using an early stopping mechanism, while 30% of the training samples are set aside as validation.

Since the data is in time-series format, it is essential to ensure the natural order of the data is kept during training, i.e., one typical error is to train on the future observations and predict the past. We split the observations into two folds as source and target using a split date. All the observations before the split date are source domain, while the observations after the split date form the target domain. Time splitting is also aligned with the development operation of the company. For example, splitting samples based on the time enables the company to take full advantage of the entire history (sensor measurements and repairs information) for each truck up to the point of deployment. We experimented on several different split dates and presented the results in Table 1. The first column gives the split dates. The second column shows the mean absolute error the network achieves, excluding the domain predictor head, i.e., only a feed-forward regressor is employed. The third column provides the results achieved by engaging both heads (domain predictor and regressor). The suggested model estimates the remaining useful life of a turbocharger. Thus, the model essentially solves a regression task. Therefore, we evaluated the model and reported the results using Mean Absolute Error metric.

To analyze the results, we can compare the mean absolute error values across each row between the second column (Source) as a baseline and the third column (Domain Adaptation). As you can see, the value of error has been reduced significantly by adding the domain adaptation head to the network. This is, of course, attained due to promoting the emergence of invariant features between source and target domains in the feature extraction layers by utilizing the domain adaptation head. Furthermore, the Domain Adaptation shows a slightly lower variance on the error rate (calculated over running the experiments five times with random network initialization).

4.3. Domain adaptation with sequential memory

The collected readouts from vehicles are time-series. Thus there is a natural sequential order available in the dataset. The method utilized in section 4.2 lacks any kind of memory and ignores this sequential information. The recurrent architecture, on the other hand, tries to address this drawback by em-

Table 2. The evaluation results comparing the RUL predictions using recurrent DANN model with two baselines.

Split	LSTM (MAE)	DANN (MAE)	DANN LSTM (MAE)
1	145.77±2.37	121±9.71	111.67±1.69
2	150.27±10.85	154.73±9.71	127.27±5.4
3	173.13±12.69	148.19±11.18	111.09±4.94
4	173.45±9.23	138.57±4.63	108.84±8.77

ploying an LSTM architecture within the feature extraction layers.

The input to the model is a sequence of 8 consecutive readouts, each containing 372 features. The number of units in the LSTM layer is 50, and it is followed by a dropout layer with a dropping frequency of 0.2. The rest of the network architecture, as well as training parameters, are the same as described in section 4.2.

Table 2 compares the results obtained by the recurrent model along with two baselines. The first baseline is called "LSTM" and uses the same architecture as recurrent, but the gradients from the domain adaptation layer are neutralized, i.e. the domain adaptation head is ignored. This baseline helps to study the effect of domain adaptation head. The second baseline is called "DANN" and uses a similar architecture as the one described previously in section 4.2. The only difference here is that the input to the network is a flattened sequence of readouts. This is done to make sure the comparison between two architectures are fair as they both receive similar input signals. This baseline helps to analyze the effect of adding sequential information (memory) to the model.

As you can see, the recurrent model has been able to improve the results one step further. The performance improvement is steady throughout the different splits. In the 2018-03-15 split as an example, the recurrent model has been able to reduce the error by approximately 21% compared to the memory-less DANN model, and 37% compared to the traditional LSTM.

4.4. Visualizing the distribution of source and target features

The whole purpose of adding domain adaptation branch is to motivate the extraction of features that are invariant in terms of source and target domains. The gradients that feature extraction layers receive from domain adaptation branch changes the internal representation of the input features in a way that makes it hard for the network to discern domains.

An interesting way of validating such an effect is to visualize the internal representation of the network. In this experiment, we visualize the internal representation produced by recurrent model. Since the dimension of the internal representation is higher than 2, we applied *t*-SNE (Van der Maaten & Hinton, 2008) to reduce the number of dimensions. For the

parameters of *t*-SNE, we set number of components equal to 2, perplexity equal to 20, and number of iterations equal to 300.

In order to study the effect of domain adaptation, we run this experiment twice. Figure 5 presents the source and target features without domain adaptation. Figure 6 represents the same features once domain adaptation is added to the model. You can see the distribution of source and target features (the red and blue points) are much more intertwined after applying domain adaptation. This indicates the network has focused more on extracting features that are invariant between two domains.

The *t*-SNE algorithm follows a stochastic process, therefore, the resulted figure changes with different seed points. However, we observed that the discussed effect is pertinent in all the experiments with different seeds.

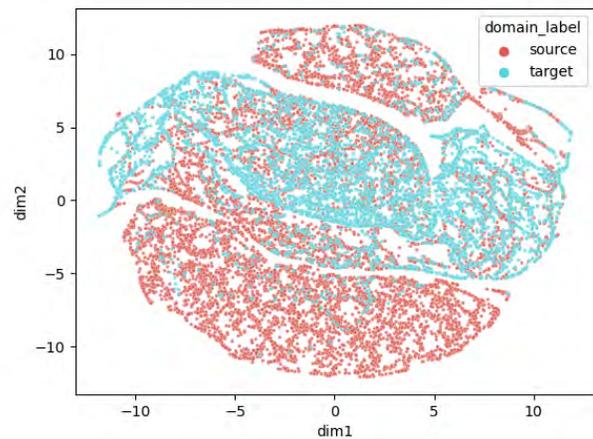


Figure 5. Internal representation of the transformed features WITHOUT domain adaptation

5. CONCLUSION

In this study, we investigated the occurrence of domain shift in the time series data from vehicle sensor readouts. The aim was to estimate the remaining useful life of turbocharger devices installed on heavy-duty trucks.

We started by showing how one can detect domain shift in the data: by fitting an autoencoder to the measurements from source and evaluating on the target distributions. Then, we employed two neural network architectures inspired from the DANN architecture. One model used a memory-less feature extractor, while the other model employed a Long Short-Term Memory unit.

Finally, we visualized how the domain adaptation changes the internal representation of input features from both source and target domains. This highlights how the adaptation is mak-

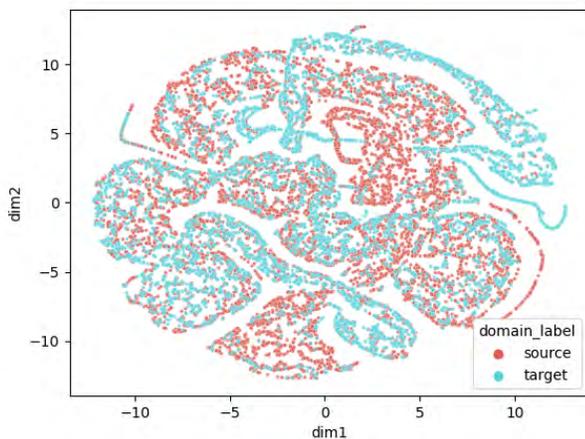


Figure 6. Internal representation of the transformed features WITH domain adaptation

ing it difficult to distinguish between the two domains. The experiments demonstrated that the domain adaptation significantly outperforms the regular architecture. We were able to improve the results even further by incorporating sequential information into the model.

The presented network architecture optimizes for two heads with different characteristics. One head is a regressor and the other one is a classifier. During the experiments, we realized the feature extraction layer receives loss values with different scales from these two heads. Tuning the scale of the losses might be helpful for balancing the effect of domain adaptation. We suggest the task's weight optimization as a future work. The current work estimates RUL of a turbocharger. However, based on each specific application, the RUL values should later be interpreted by the user and considered for decision making. This could also be affected for example by the scheduled workshop visits for each truck (if we want to do the maintenance in the pre-booked regular maintenance visits). The decision making process based on the RUL values is left as a future work.

ACKNOWLEDGMENT

The authors would like to thank VOLVO Trucks Corporation for providing access to the data used in this study. The work was supported by grants from KK-Foundation and Vinnova.

REFERENCES

Altarabichi, M. G., Fan, Y., Pashami, S., Mashhadi, P. S., & Nowaczyk, S. (2021). Extracting invariant features for predicting state of health of batteries in hybrid energy buses. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 1–6).

Che, Y., Deng, Z., Lin, X., Hu, L., & Hu, X. (2021). Predictive battery health management with transfer learning and online model correction. *IEEE Transactions on Vehicular Technology*, *70*(2), 1269–1277.

da Costa, P. R. d. O., Akçay, A., Zhang, Y., & Kaymak, U. (2020). Remaining useful lifetime prediction via deep domain adaptation. *Reliability Engineering & System Safety*, *195*, 106682.

Ding, Y., Jia, M., Miao, Q., & Huang, P. (2021). Remaining useful life estimation using deep metric transfer learning for kernel regression. *Reliability Engineering & System Safety*, *212*, 107583.

Fan, Y., Nowaczyk, S., & Rögnvaldsson, T. (2015a). Evaluation of self-organized approach for predicting compressor faults in a city bus fleet. *Procedia Computer Science*, *53*, 447–456.

Fan, Y., Nowaczyk, S., & Rögnvaldsson, T. S. (2015b). Incorporating expert knowledge into a self-organized approach for predicting compressor faults in a city bus fleet. In *Scai* (pp. 58–67).

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The journal of machine learning research*, *17*(1), 2096–2030.

Kharazian, Z., Rahat, M., Fatemizadeh, E., & Nasrabadi, A. M. (2020). Increasing safety at smart elderly homes by human fall detection from video using transfer learning approaches. In *30th European Safety and Reliability Conference (ESREL2020) & 15th Probabilistic Safety Assessment and Management Conference (PSAM15), Venice, Italy, 1-5 November, 2020*.

Li, F., Tang, T., Tang, B., & He, Q. (2021). Deep convolution domain-adversarial transfer learning for fault diagnosis of rolling bearings. *Measurement*, *169*, 108339.

Mao, W., Liu, Y., Ding, L., Safian, A., & Liang, X. (2020). A new structured domain adversarial neural network for transfer fault diagnosis of rolling bearings under different working conditions. *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–13.

Mashhadi, P. S., Nowaczyk, S., & Pashami, S. (2020). Stacked ensemble of recurrent neural networks for predicting turbocharger remaining useful life. *Applied Sciences*, *10*(1), 69.

Prytz, Nowaczyk, S., Rögnvaldsson, T., & Bytner, S. (2015). Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data. *Engineering applications of artificial intelligence*, *41*, 139–150.

Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (2008). *Dataset shift in machine learning*. Mit Press.

Rahat, M., Pashami, S., Nowaczyk, S., & Kharazian, Z. (2020). Modeling turbocharger failures using markov process for predictive maintenance. In *30th European*

safety and reliability conference (esrel2020) & 15th probabilistic safety assessment and management conference (psam15), venice, italy, 1-5 november, 2020.

- Revanur, V., Ayibiowu, A., Rahat, M., & Khoshkangini, R. (2020). Embeddings based parallel stacked autoencoder approach for dimensionality reduction and predictive maintenance of vehicles. In *Iot streams for data-driven predictive maintenance and iot, edge, and mobile for embedded machine learning* (pp. 127–141). Springer.
- Uddagiri, V. S. V., Ramalingam, S. N. B., Rahat, M., & Mashhadi, P. S. (2021). Predicting hybrid vehicles' fuel and electric consumption using multitask learning. In *2021 IEEE 8th international conference on data science and advanced analytics (dsaa)* (pp. 1–6).
- Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).
- Wang, J., Liang, Y., Zheng, Y., Gao, R. X., & Zhang, F. (2020). An integrated fault diagnosis and prognosis approach for predictive maintenance of wind turbine bearing with limited samples. *Renewable Energy*, 145, 642–650.
- Xia, T., Song, Y., Zheng, Y., Pan, E., & Xi, L. (2020). An ensemble framework based on convolutional bi-directional lstm with multiple time windows for remaining useful life estimation. *Computers in Industry*, 115, 103182.
- Yang, M., Zhao, W., Chen, L., Qu, Q., Zhao, Z., & Shen, Y. (2019). Investigating the transferring capability of capsule networks for text classification. *Neural Networks*, 118, 247–261.
- Zhang, W., Yang, D., & Wang, H. (2019). Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal*, 13(3), 2213–2227.
- Zhang, Y., Hutchinson, P., Lieven, N. A., & Nunez-Yanez, J. (2020). Remaining useful life estimation using long short-term memory neural networks and deep fusion. *IEEE Access*, 8, 19033–19045.

Experimental assessment of a broadband vibration and acoustic emission sensor for rotorcraft transmission monitoring

Cristobal Ruiz-Carcel¹, Andrew Starr¹, and Arturo Francese¹

¹ *Cranfield University, Cranfield, Bedfordshire MK430AL, United Kingdom*

c.ruizcarcel@cranfield.ac.uk

a.starr@cranfield.ac.uk

a.francese@cranfield.ac.uk

ABSTRACT

Modern rotorcrafts rely on Health and Usage Monitoring Systems (HUMS) to enhance their availability, reliability, and safety. In those systems, data related to the health of key mechanical components is acquired, in addition to typical flight condition history data such as speed and torque. Commercial HUM systems usually rely on vibration measurements to assess the condition of shafts, gears, and bearings; using techniques such as spectral analysis, harmonic analysis, vibration trend and others. Recent research has shown that acoustic emissions (AE) can be advantageous in the detection of mechanical faults, in particular detecting very early small defects on bearings and gears, providing extra time for maintenance planning. However, the addition of extra sensors adds complexity and weight to the HUMS system, which is undesirable. This research is an experimental study to assess the monitoring capabilities of a broadband sensor, able to cover both low frequency vibration components as well as ultrasonic events, hence combining the benefits of both in a single compact sensing unit. The experimental results obtained from an instrumented rig using healthy components as well as seeded faults show the ability of the sensor to detect high frequency events, and compares the performance of the sensor in the low frequency range with a commercial accelerometer.

1. INTRODUCTION

Health and Usage Monitoring Systems (HUMS) are used to monitor rotorcraft power transmission systems, typically using predefined vibration features to assess their condition (Decker, 2002; Zakrajsek et al., 1993, 1995). HUMS was originally developed in North Sea operations, especially after the accident of a Boeing-Vertol 234 in 1986 caused by a main gearbox failure.

HUMS have two main functions, health monitoring and usage monitoring. The first aims to diagnose mechanical damage in the very early stages of degradation, before it leads to catastrophic damage. Usage monitoring focuses on the assessment of operation hours, current components condition and load history to estimate remaining life of mechanical components (Decker & Lewicki, 2003; Samuel & Pines, 2005). Commercial HUMS make use of different vibration analysis methods to detect faults in bearings, gears and shafts. Condition Indicators (CI's) are key vibration features extracted from the acquired vibration signals, which can be related to specific mechanical faults (Dempsey et al., 2008). In HUMS a range of different CI's are extracted from vibration data to characterize component health.

Vibration analysis has been traditionally grouped in three main categories; time domain, frequency domain and time-frequency domain. Time domain analysis pre-processes the raw signals (if necessary) and extracts features such as rms, skewness, and kurtosis (Martin, 1989; Sait & Sharaf-Eldeen, 2011). The Fast Fourier transform (FFT) is commonly used to obtain the frequency spectra of the signals, revealing their fundamental components. Fault detection in the frequency domain is based on identification of certain frequencies associated with bearing or gear faults. The amplitude of the components associated to those frequencies is then used as a CI. Time-frequency domain methods are able to track changes in the signal composition over time, including techniques such as short-time Fourier transform (STFT) (Mehala & Dahiya, 2008), Wigner-Ville (Sait & Sharaf-Eldeen, 2011), and wavelet analysis (Wang & McFadden, 2010).

Acoustic emissions (AE) in the field of machine monitoring are defined as transient elastic waves produced by the interface of two components or more in relative motion (Mba & Rao, 2006). Typical AE sources include impacts, crack growth, friction, turbulence, material loss, cavitation, leakage etc. Its main benefit against vibration analysis and oil analysis is the capability to detect faults earlier due to the high sensitivity offered by AE (Tan et al., 2007). On the other

Cristobal Ruiz-Carcel et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

hand, the main drawback of AE is the difficulty in processing, interpreting and manipulating the acquired data (Al-Ghamd & Mba, 2006; Couturier & Mba, 2008). In addition, AE waves suffer a rapid attenuation of the signal, and require the AE sensor to be close to the source.

Vibration-based gearbox monitoring is well established, however the application of AE to this field is still in its early stages (Y. Qu et al., 2013; Tan et al., 2007) and it is difficult to see it implemented in commercial tools. In the area of HUMS some research has been carried out in recent years to prove the capabilities of AE to monitor helicopter transmission components, focusing on epicyclic gearboxes (Duan et al., 2015; Elasha et al., 2017; A. Qu et al., 2013). These investigations concluded that AE offered much earlier indication of damage than vibration analysis, and the proposed processing techniques were suitable for gearbox fault diagnosis.

Helicopter transmission systems are quite complex and compact, with difficult access and a requirement for lightweight. Hence it is necessary to simplify the monitoring system as much as possible, minimizing the number of sensors and wiring to reduce weight and requirements for sensor installation. The research in this paper assesses the capabilities of a broadband acoustic emission sensor, with a frequency range of 0.1 Hz to 1MHz, as a unique AE and vibration sensing unit for helicopter gearbox monitoring. Although the theoretical frequency range covered by the sensor supports its suitability as a vibration sensor as well as an AE sensor, in practice it is extremely difficult to build a sensor with a flat frequency response in such a wide range, which could hinder fault detection based on traditional vibration analysis. Consequently, the objective of this research is to compare the monitoring capabilities of this sensor with a commercial accelerometer, based on analysis of signals obtained on a laboratory scale rig where faults were artificially introduced. After signal amplification and digitation, the signal is high and low pass filtered to divide the AE and vibration content in it, which are analyzed separately. The main benefit of such approach is the simplification of the sensing unit, minimizing weight and required space, while maintaining the benefits of vibration and AE monitoring simultaneously.

2. METHODOLOGY

2.1. Sensing

The iMPactXS high-performance acoustic emission and dynamic load sensor manufactured by iNDTact GmbH was selected due to its frequency range (0.1 Hz to 1MHz) and sensitivity (> 1200 pC/N). As shown in Figure 1, this sensor covers the typical frequency ranges of both, vibration and AE sensors. The sensors were connected to an iNDTact champ charge amplifier, and the signals were digitized at 2 MHz

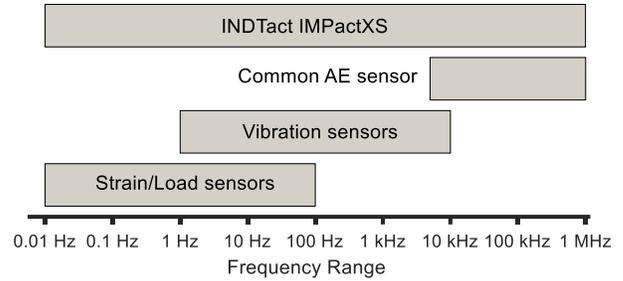


Figure 1: Typical frequency band of different sensors (iNDTact GmbH, 2022)

using a Pico Technology PicoScope 4224 IEPE digital oscilloscope.

Accelerometers are characterized by a flat frequency response in their usable frequency range, typically up to 10 or 20 kHz. That characteristic allows a direct conversion of the sensor output signal in mV to acceleration in ms^{-2} , as the sensitivity is constant in that limited frequency region. In order to assess the vibration monitoring capabilities of the broadband sensor, a commercial accelerometer will also be used simultaneously during the tests. A triaxial accelerometer (Brüel & Kjaer 4535-B) with a frequency range of 0.3 to 10000 Hz in the X and Y axes, and 0.3 to 12800 Hz in Z was selected. The voltage sensitivity is 1 mV/ ms^{-2} . Vibration signals were sampled at 51.2 kHz using a National Instruments 9234C data acquisition card.

In order to ensure that the transmission path for both sensors is equivalent a special sensor cluster was designed. Both sensors were installed in a compact machined aluminum block as shown in Figure 2. The AE sensor was glued using Dow Corning 3140 as a wave couplant, whereas the AE sensor was bolted to the metal block with an M3 stud. The cluster was attached to the rig using Loctite EA 9492.



Figure 2: Sensor cluster detail

2.2. Signal processing strategy

The signal processing strategy used for both sensors is represented in the diagram in Figure 3. Once digitized the vibration signals are processed directly for feature extraction. The AE signals however, after preamplification and digitation are divided in two categories using digital low and high pass filters with a cut-off frequency of 20 and 70 kHz respectively. Such approach allows for individual analysis of “low frequency” vibration-like events (such as oscillations

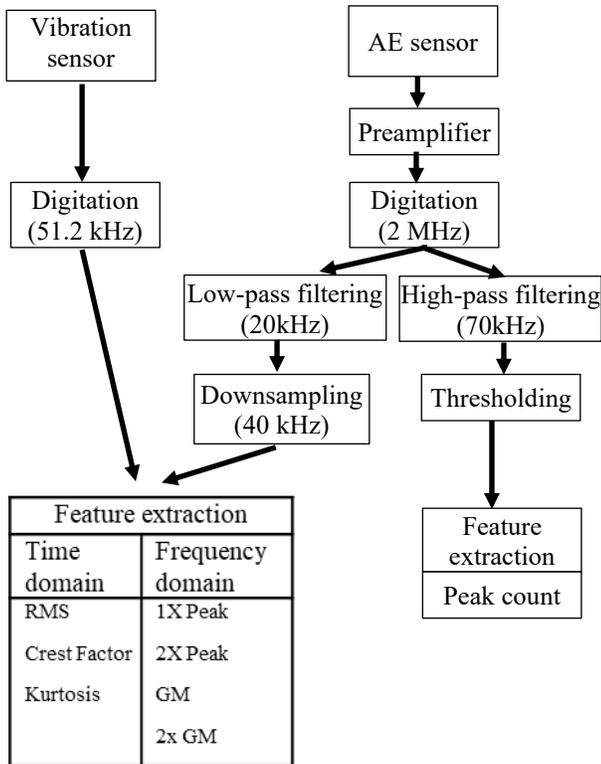


Figure 3: Data analysis workflow

related to misalignment, unbalance, gear mesh, bearing faults or resonances) independently from “high frequency” events such as impacts, friction or crack growth. The low frequency part of the signal is then down sampled to 40 kHz, which is enough to accurately represent the vibration signature and reduces computational cost. The features extracted from vibration signals in the time domain are RMS, crest factor and Kurtosis. The frequency spectrum is obtained using the FFT, and the peak amplitudes at the input shaft frequency (1X), the second harmonic (2X), the gear mesh frequency (GM) and its harmonic (2xGM) are extracted. This collection of CI’s is equal for both, the vibration sensor and the low frequency part of the broadband sensor. The selected CI’s are basic, well understood and widely used in gear monitoring, making them appropriate for this comparison exercise. The



Figure 4: Gearbox rig

high frequency part is analysed by setting a threshold above

the background noise level (thresholding), and counting the number of occasions the signal exceeded that threshold.

2.3. Experimental setup

The rig used to assess the performance of the sensors is a single stage gearbox rig (Figure 4), powered by an 11kW induction motor with 2 pairs of poles, and a nominal speed of 1490 rpm. The output shaft is connected to a dynamometer that absorbs and measures the load applied. The gear pair has straight teeth, a module of 5, and 24 and 25 teeth in the input and output shafts respectively. A lubrication port on the gearbox casing cover provided lubrication from an external pump. Although this benchtop arrangement is quite different from a helicopter gearbox in terms of shape, size, power, and stiffness, the gear meshing dynamics are the same as in any gear pair. The transmission path for the fault generated forces through the gears, shafts, and bearings to the static components are also comparable. Figure 5 (left) shows a detail of the gear pair. Figure 5 (right) shows the location of the sensor cluster, installed on a flat surface in the vicinity of the input shaft bearing housing.



Figure 5: Gear pair (left) and sensor cluster location (right)

In addition to the AE sensor and the triaxial accelerometer, the rig is equipped with a shaft speed sensor and a torque sensor. Temperature of the sensor cluster was also measured using a thermocouple.

2.4. Testing procedure

Data from both sensors, as well as speed, torque and temperature measurements were acquired during testing. The dynamometer was set to four different torque setpoints (10, 20, 30, and 40 Nm) to assess the sensor response at different loads. Vibration data was acquired in recordings of 1s, while the broadband sensor data recordings lasted 0.2s in order to keep a reasonable volume of data due to high sampling rate.

Initially the rig was operated with healthy gears to set a baseline for all the CI’s studied. Spalling was artificially introduced in the contact surface of one of tooth in the diver side gear to study the evolution of the CI’s. This failure mode was selected as planetary gear sets are more vulnerable to pitting defects due to intricate lubrication conditions. With the increase of the running cycles, the micro pitting will

induce more deleterious faults, such as spalling and chipping (Huangfu et al., 2022) Three different defect sizes were tested, drilling holes on the gear tooth surface of 0.8 (small), 1 (medium) and 1.5 (large) mm of diameter and around 0.2 mm in depth (Figure 6).



Figure 6: Detail of artificially introduced spalling

3. RESULTS AND DISCUSSION

3.1. Signals Overview

3.1.1. High frequency/AE

In first place the high-pass filtered broadband sensor signal was observed to assess the capabilities of the broadband sensor to detect small defects. Figure 7 shows a sample of the signals acquired for the healthy case (top) and the small fault (bottom) under 30 Nm of load. The healthy signal is composed basically by background noise, and there are no obvious bursts or peaks in the signal that indicate detection of AE related events. The faulty case however shows a series of bursts that clearly stand out of the carpet level. In addition, the distance between those bursts is around 40 ms, which is the time it takes for a full input shaft rotation to occur. That is the rate at which the induced fault enters the gear mesh.

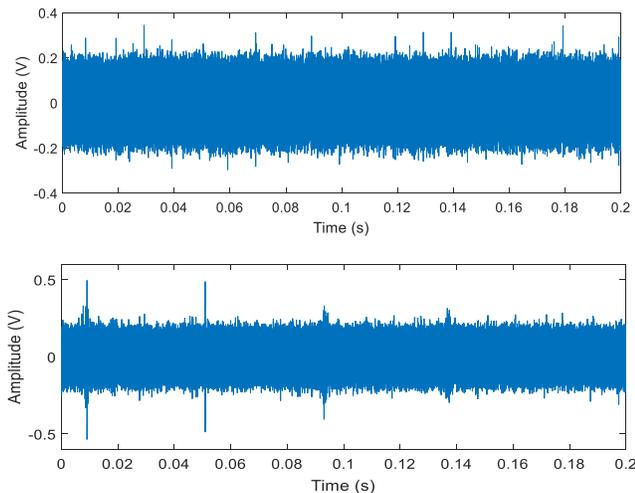


Figure 7: AE signal sample, healthy (top) and small fault (bottom) at 30 Nm

3.1.2. Low frequency/Vibration

For the comparison between the accelerometer and the low-frequency part of the broadband sensor, only the Z direction of the accelerometer (perpendicular to the mounting surface) was considered for simplicity, as it is the same direction the broadband sensor is measuring. Figure 8 shows an example of the frequency content from the signal acquired from both sensors in the healthy case and 30 Nm of load. It can be seen that despite the lack of faults the spectrum is dominated by the GM frequency and its harmonics, as usual in gearboxes. The spectrum also shows that these main peaks are surrounded by sidebands, spaced around 24 Hz (the input shaft frequency) from each other. That may be an indication that the alignment between the shafts is not perfect, and the gear mesh is being modulated in amplitude once per revolution.

When comparing the spectrum of both sensors, it can be seen that the frequency content of the broadband sensors is similar to the accelerometer, but the relative amplitude of the peaks differs. It is important to note that the units have been kept in V for both sensors and each one has its own scale, as the potential lack of linearity in the response of the broadband sensor does not allow a direct conversion to acceleration units. The results at the top graph in Figure 8 show that the broadband sensor is able to capture the most relevant components, GM and harmonics, but fails to accurately

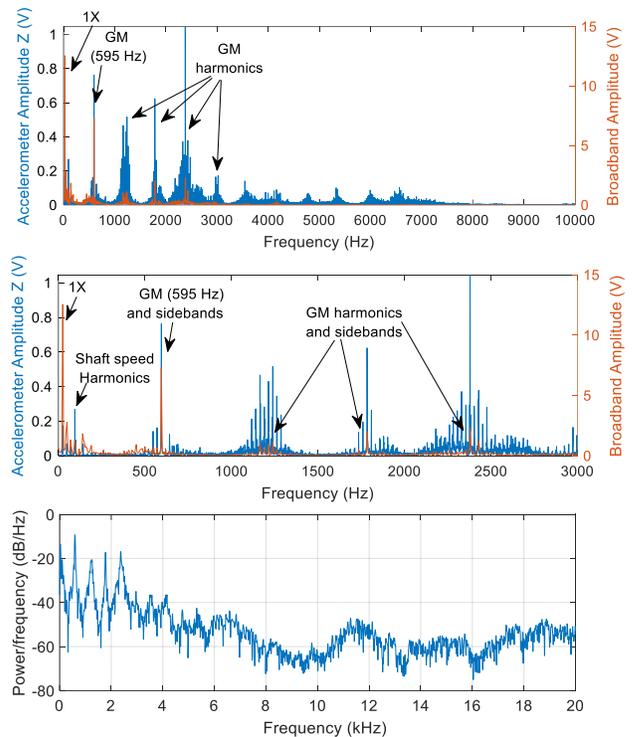


Figure 8: Frequency spectrum of accelerometer and broadband sensor signals up to 10 kHz (top), low frequency detail (centre) and Welch cross power spectral density estimate (bottom)

Table 1: Number of samples obtained in each case

Case	Healthy				Small				Medium				Large			
Torque (Nm)	10	20	30	40	10	20	30	40	10	20	30	40	10	20	30	40
Sample number	189	165	184	177	72	59	82	65	92	73	95	73	123	127	119	117

capture the amplitude of signal components beyond 3000 Hz. This observation is also corroborated by the Welch cross power spectral density estimate presented in Figure 8 (bottom). This analysis shows the highest coherence between the signals is found below 3 kHz and is particularly high for the GM frequency and its first harmonics. For higher frequencies the coherence is much lower. In principle that is not a huge problem, as the most typical vibration signatures in rotating machinery (shaft speed, GM, bearing defect frequencies, etc. and harmonics of all of them) typically happen in that region. However other important vibration phenomena, particularly resonances, typically happen between 3 and 10 kHz, which could be a problem for this sensor. On the other hand, the AE capabilities should be able to capture impact-like events even earlier than an accelerometer could detect a change in the amplitude of a resonant frequency.

The lack of linearity in the frequency response compared with the accelerometer is quite evident when looking closely at the amplitudes of the main components (Figure 8 centre). Even if the 1X peak amplitude and most of its harmonics are larger than the same peaks in the accelerometer signal spectrum with the selected axes ranges, the correlation is not maintained for higher frequency components (mainly GM, harmonics and sidebands). It can be concluded that the broadband sensor is not as good at responding linearly to a range of different frequencies as the accelerometer. This would be a problem for approaches where it is required to obtain an accurate measurement of acceleration at different frequencies. Commonly that is not the case in monitoring applications, where the typical procedure is to compare newly acquired measurements with a established baseline. From that perspective, repeatability and precision in the representation of amplitude for different frequency components are way more important than accuracy. As it will be seen later, repeatability in measured peak amplitudes was not an issue for this sensor. Despite the lack of fidelity in amplitude compared with the accelerometer, the broadband sensor was able to accurately identify the main frequency components in the signal, which is key to identify the sources of vibration and possible links to mechanical faults. Consequently, the signals acquired from the sensor are in principle adequate and acceptable for monitoring purposes. The next subsection will investigate fault detection performance.

3.2. Fault detection

Table 1 shows the number of samples obtained for each combination of torque and healthy/faulty case. The results will be presented displaying the average value of the CI

obtained for each combination, and the standard deviation of each sample will be represented in the form or an error bar around the mean value.

3.2.1. High frequency/AE

Presence of peaks in the AE signals for the healthy case should not happen, as only events related to faults produce AE activity. For the faulty cases peak count will depend on rotational speed, defect frequency and whether the amplitude of the AE generate dis large enough to cross the threshold. Preliminary analysis of the AE signals obtained under healthy conditions, revealed that the maximum absolute value observed was on average 0.27 V (seen example in Figure 7 top). Consequently the threshold value above that carpet level was set to 0.3 V, which provided the peak counting results shown in Figure 9. As it can be seen, the number of threshold crossings found in the healthy case is small for all loads, and can be attributed to outliers slightly above the threshold level, which was chosen relatively low to enhance sensitivity. The effect of the fault was evident even in the small fault case, particularly for high loads. This result highlights the main benefit of AE and its capability to identify faults in the early stages of degradation. The medium and large fault cases also show larger number of threshold crossings than the healthy case, and the standard deviation in the samples is greater particularly in the large case.

3.2.2. Low frequency/Vibration

Figure 10 shows the results obtained from the time domain vibration CI's extracted from the accelerometer and the broadband sensor. RMS shows little sensitivity to the fault in the small and medium cases, and it was only in the large case that an increment in this indicator was obvious. The case of the CF is not very informative, as the changes in the mean values observed are in the same order of magnitude as the standard deviations. No significant differences with the healthy case are observed. Important to note that the CF is

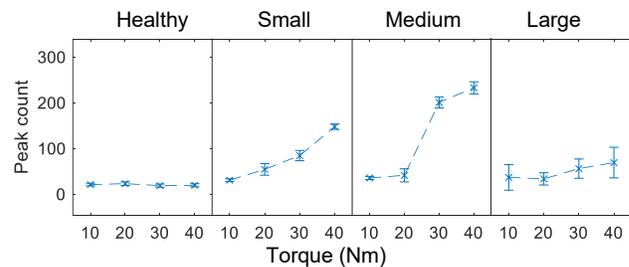


Figure 9: Average value and standard deviation for peak count from AE signal

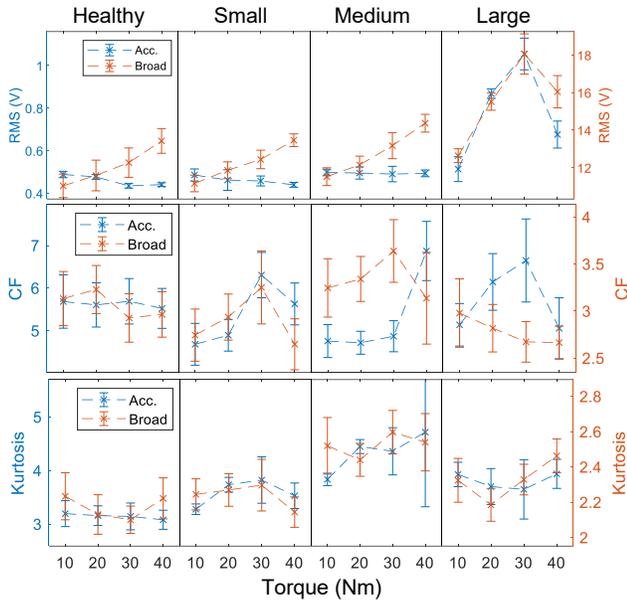


Figure 10: Average value and standard deviation for time domain CI's

typically twice as large for the accelerometer than for the broadband sensor, evidencing a larger signal to noise ratio. Kurtosis does not show any evident failure indications for the small case, but the values in the medium case are significantly larger than the healthy case, although they decay again for the large fault case. K values are smaller for the broadband sensor, pointing at a lack of “peakiness” in the signals compared with the accelerometer. It is important to note that despite the differences in magnitude between both sensors for all three indicators, their response to changes in load and presence of faults is similar.

The results obtained from frequency domain CI's are presented in Figure 11. The amplitude of the shaft speed peak (1X) shows no significant change with the small fault compared with the healthy case. However in both the medium and large cases there is an increment in the CI in both sensors, which is also appreciable as the load increases. The second harmonic of the shaft speed does not present significant changes in the presence of faults for any of the sensors, and any variations are in the same order of magnitude as the standard deviation for the healthy case. The GM peak amplitude shows a small increment for the medium and large faults, but not noticeable increments for the small fault. This CI is clearly correlated with load as well. Its second harmonic shows a very similar behaviour to the 2X case with even greater variability in the accelerometer measurements in the faulty cases. The broadband sensor shows some increment with respect to the healthy case in the medium and large fault cases, but again variability in this CI is too large to consider the differences significant.

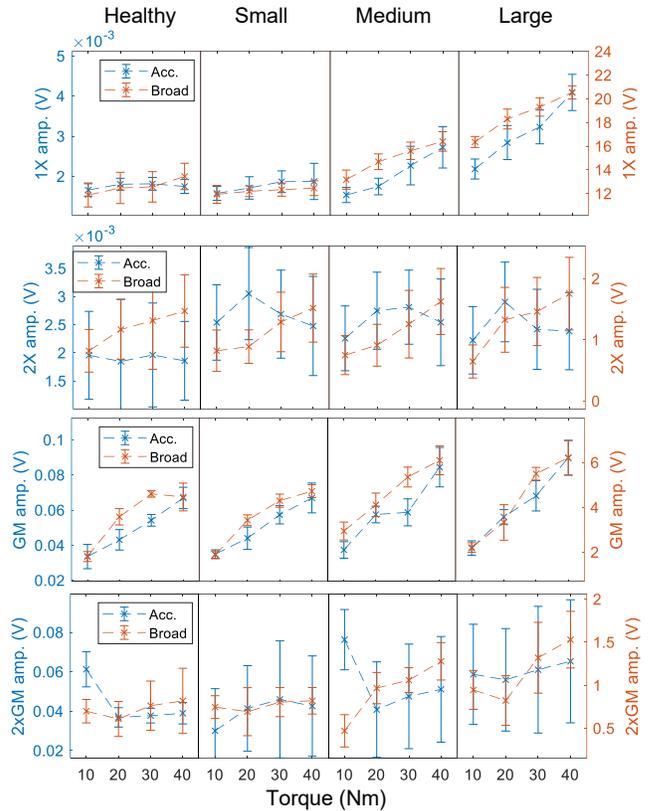


Figure 11: Average value and standard deviation for frequency domain CI's

Table 2 summarises the results presented in a tabular format, displaying the percentage change with respect to the healthy case of every CI considered in both sensors. It is quite clear that AE based peak counting was the only indicator providing a clear increment for the small fault case, as seen before the 2X peak amplitude measurements in the accelerometer had large variability and hence low significance. The 1X and GM peak amplitudes, together with Kurtosis were able to react to the medium and large cases. RMS was only sensitive to the large fault. Those results are quite consistent for both sensors.

4. CONCLUSION

The research presented in this paper focuses on an experimental validation of a broadband sensor for rotorcraft transmission monitoring, which combines AE and vibration monitoring capabilities in a single sensor. The validation was performed through comparison of the selected sensor with a conventional accelerometer, which were both tested on a dedicated gear rig operated at different loads where gear spalling at 3 different degradation stages was introduced artificially. The proposed data processing technique separates the broadband sensor signal in its low and high frequency regions, allowing the use of traditional vibration and AE analysis techniques to be applied for feature extraction.

Table 2: Mean value percent change with respect to the corresponding healthy case

Case	Large				Medium				Small			
	10	20	30	40	10	20	30	40	10	20	30	40
Torque												
RMS acc.	-0.5	-3.1	5.2	-0.2	1.9	3.8	12.7	12.0	5.1	82.1	142.6	53.5
RMS broad.	1.1	2.4	1.4	0.3	4.4	4.9	7.4	7.0	14.6	33.9	47.2	19.5
CF acc.	-17.6	-12.7	10.8	1.9	-16.3	-15.8	-14.4	24.4	-9.6	9.6	16.8	-8.4
CF broad.	-12.4	-9.1	11.1	-10.7	3.6	3.5	24.3	5.8	-4.9	-12.7	-8.7	-10.1
K acc.	2.6	8.8	2.5	1.5	20.0	40.8	38.7	53.0	22.9	17.2	16.0	27.8
K broad.	0.5	6.5	9.3	-3.5	12.8	14.5	23.6	14.3	4.0	2.6	10.8	10.8
1X acc.	-5.1	-5.1	2.9	7.2	-7.4	-3.2	25.0	55.6	31.4	57.4	78.0	133.8
1X broad.	0.4	-2.2	-1.9	-7.4	10.9	17.7	24.2	22.0	37.6	46.6	53.6	52.8
2X acc.	29.8	65.3	37.0	33.3	15.4	48.8	43.3	36.8	13.6	57.3	23.3	28.4
2X broad.	0.1	-23.9	-1.8	3.2	-8.7	-22.0	-4.5	10.5	-21.2	13.5	11.2	19.3
GM acc.	4.2	2.1	5.8	0.0	11.2	33.4	8.4	26.2	14.5	30.3	25.7	34.2
GM broad.	4.2	-4.3	-6.6	5.8	63.1	14.9	16.3	36.6	22.5	-7.0	19.5	39.2
2xGM acc.	-50.9	12.2	21.8	9.3	24.6	10.7	25.9	31.1	-4.1	51.6	61.7	67.6
2xGM broad.	6.8	13.9	5.4	0.1	-32.6	58.8	38.8	56.0	35.3	35.1	72.9	86.7
AE Peak count	45.6	135.5	349.0	656.5	68.5	79.0	967.6	1091.4	74.8	45.3	197.2	254.0

The results obtained showed that the high frequency analysis of the signal was able to detect the smallest fault introduced, proving its capability to provide early fault detection as expected from an AE sensor. The analysis of the spectrum for the low frequency part of the signal showed that the broadband sensor can identify the same signal components measured by the accelerometer, which are related to the operating conditions, the machine's components geometry and their condition. However, the relative amplitudes of those components were different to the observations in the accelerometer, pointing to a lack of linearity in the frequency response of the sensor in the frequency range where vibration components are typically manifested. Even though, the main components were still easily identifiable, and the amplitude measured was repeatable throughout the tests, proving its ability to consistently provide a reliable comparison with a baseline value. The amplitude of all components over 3 kHz was clearly diminished, which can be a problem for approaches that require the study of frequencies in this range, such as resonances.

The CI's extracted from both sensors showed a very similar response to changes in load and presence of faults, proving that the sensor is suitable for vibration monitoring based on analysis of basic vibration features. None of the vibration features studied provided a clear increment for the smallest fault studied that was statistically significant. For time domain analysis, RMS was only sensitive to the large fault, while Kurtosis showed some indication of change for the medium case. In frequency domain analysis, both the 1X and the GM CI's increased for the medium and particularly for the large case, proving effective in detecting the fault. Variability was too large in the analysis of CF, 2X and 2xGM. Future work will need to investigate if the similarities between the CI's from both sensors can be extended to more

complex vibration analysis techniques, such as bispectral analysis or cyclo-stationary analysis.

The most important conclusion from the analysis of this range of CI's was that the behaviour of both sensors was very similar (despite the differences in scale), proving that the capabilities of the broadband sensor for vibration-based monitoring are comparable to the accelerometer. Hence a monitoring system based solely on this sensor could combine the benefits of both, AE and vibration monitoring with a single sensing unit and combined data processing.

ACKNOWLEDGEMENT

This research has received funding from the Clean Sky 2 Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 738144.

REFERENCES

Al-Ghamd, A. M., & Mba, D. (2006). A comparative experimental study on the use of acoustic emission and vibration analysis for bearing defect identification and estimation of defect size. *Mechanical Systems and Signal Processing*, 20(7), 1537–1571. <https://doi.org/10.1016/j.ymssp.2004.10.013>

Couturier, J., & Mba, D. (2008). Operational bearing parameters and acoustic emission generation. *Journal of Vibration and Acoustics, Transactions of the ASME*, 130(2). <http://www.scopus.com/inward/record.url?eid=2-s2.0-45249088282&partnerID=40&md5=a193d74487821fb420a9e45cdfc1a6fd>

Decker, H. (2002). *Crack Detection for Aerospace Quality Spur Gears*. 15.

- Decker, H., & Lewicki, D. (2003). *Spiral Bevel Pinion Crack Detection in a Helicopter Gearbox*. 16.
- Dempsey, P., Keller, J., Wade, D., & Arsenal, R. (2008). *Signal Detection Theory Applied to Helicopter Transmission Diagnostic Thresholds*.
- Duan, F., Elasha, F., Greaves, M., & Mba, D. (2015). *Helicopter Main Gearbox Bearing Defect Identification using Vibration and Acoustic Emission Techniques*.
<https://doi.org/10.1109/ICPHM.2016.7542856>
- Elasha, F., Greaves, M., Mba, D., & Fang, D. (2017). A comparative study of the effectiveness of vibration and acoustic emission in diagnosing a defective bearing in a planetary gearbox. *Applied Acoustics*, 115, 181–195.
<https://doi.org/10.1016/J.APACOUST.2016.07.026>
- Huangfu, Y., Dong, X., Chen, K., Tu, G., Long, X., & Peng, Z. (2022). A tribo-dynamic based pitting evolution model of planetary gear sets: A topographical updating approach. *International Journal of Mechanical Sciences*, 220, 107157.
<https://doi.org/10.1016/J.IJMECSCI.2022.107157>
- iNDTact GmbH. (2022, March 25). *Datasheet iMPactXS*.
https://www.indtact.de/files/Ugd/5a81ff_463278d4e34446fabca5b70adf13c592.Pdf
- Martin, H. (1989). Statistical Moment Analysis as a Means of Surface Damage Detection. *Proceedings of the International Modal Analysis Conference*.
- Mba, D., & Rao, R. B. K. N. (2006). Development of acoustic emission technology for condition monitoring and diagnosis of rotating machines: Bearings, pumps, gearboxes, engines, and rotating structures. *Shock and Vibration Digest*, 38(2), 3–16.
<http://www.scopus.com/inward/record.url?eid=2-s2.0-32944479589&partnerID=40&md5=16e5b54a47634aaca6788c18bb5fad9f>
- Mehala, N., & Dahiya, R. (2008). *A comparative study of FFT, STFT and wavelet techniques for induction machine fault diagnostic analysis*.
- Qu, A., Hecke, B., He, D., Yoon, J., Bechhoefer, E., & Zhu, J. (2013). Gearbox Fault Diagnostics using AE Sensors with Low Sampling Rate. *Journal of Acoustic Emission*, 31, 67–90.
- Qu, Y., Hecke, B. van, He, D., & Yoon, J. (2013). *Gearbox Fault Diagnostics using AE Sensors with Low Sampling Rate*. 31, 67–90.
- Sait, A., & Sharaf-Eldeen, Y. (2011). A Review of Gearbox Condition Monitoring Based on vibration Analysis Techniques Diagnostics and Prognostics. In *Conference Proceedings of the Society for Experimental Mechanics Series* (Vol. 5, pp. 307–324).
https://doi.org/10.1007/978-1-4419-9428-8_25
- Samuel, P., & Pines, D. (2005). A review of vibration-based techniques for helicopter transmission diagnostics. *Journal of Sound and Vibration*, 282, 475–508.
<https://doi.org/10.1016/j.jsv.2004.02.058>
- Tan, C. K., Irving, P., & Mba, D. (2007). A comparative experimental study on the diagnostic and prognostic capabilities of acoustics emission, vibration and spectrometric oil analysis for spur gears. *Mechanical Systems and Signal Processing*, 21(1), 208–233.
<https://doi.org/10.1016/J.YMSSP.2005.09.015>
- Wang, W. W., & McFadden, P. D. (2010). Application of wavelets to gearbox vibration signal for fault detection. *Journal of Sound and Vibration*, 192, 927–939.
<https://doi.org/10.1006/jsvi.1996.0226>
- Zakrajsek, J., Townsend, D., & Decker, H. (1993). *An Analysis of Gear Fault Detection Methods as Applied to Pitting Fatigue Failure Data*.
- Zakrajsek, J., Townsend, D., Lewicki, D., Decker, H., & Handschuh, R. (1995). *Transmission Diagnostic Research at NASA Lewis Research Center*. 14.

BIOGRAPHIES

Dr. Cristobal Ruiz-Carcel Cristobal received his degree in mechanical engineering in 2010 from Universidad Politecnica de Valencia (Spain), followed by an MSc Eng. degree in Design of Rotating Machines from Cranfield University in 2011. In 2014 he completed his PhD in the field of condition monitoring applied to large scale industrial systems at Cranfield University, which was part of the Marie Curie FP7 project “Energy-Smartops”. Cristobal's main research interests have been focused on signal processing algorithms, multivariate data analysis, condition monitoring and predictive maintenance. He is currently a Research Assistant at the Centre for Life-cycle Engineering and Management, where he works on the development, testing, and implementation of novel monitoring techniques for different applications.

Prof. Andrew Starr is head of the Centre for Life-cycle Engineering and Management (CLEM), a world-leading centre of excellence in maintenance and asset management for high value systems. The centre works in partnership with industry in research and education. Professor Starr read Mechanical Engineering at the University of Leeds, while sponsored by British Aerospace (Civil Aircraft), for which he was awarded first prize in the final year of his apprenticeship. He studied for his doctoral thesis in condition based maintenance for robotic production plant at the University of Manchester, sponsored by Ford and Wolfson Maintenance. He has held academic posts at the University of Huddersfield, the University of Manchester, and the University of Hertfordshire, as Head of the School of Aerospace, Automotive and Design Engineering. He has published over 150 technical papers from a wide range of collaborative projects with industry, helping to solve real problems and to devise innovative products and services.

Arturo Francese is a chartered engineer working as senior stress engineer at Cranfield Aerospace Solution ltd. He has more than 20 years of experience in the aerospace industry, mainly in the stress department and in support to production team. During his career, he has been a signatory holder for concessions and design changes to legacy products. Currently he is involved with Cranfield University by supporting research projects and taught students.

Optical Cutting Tool Wear Monitoring by 3D Geometry Reconstruction

Rob Salaets¹, Valentin Sturm², Ted Ooijevaar³, Veronika Putz⁴, Julia Mayer⁵ and Abdellatif Bey-Temsamani⁶

^{1,3,6} *Flanders Make vzw, CoreLab DecisionS, Leuven, 3001, Belgium*
rob.salaets@flandersmake.be
ted.ooijevaar@flandersmake.be
abdellatif.bey-temsamani@flandersmake.be

^{2,4,5} *Linz Center of Mechatronics GmbH, Sensors & Communication, 4040 Linz, Austria*
valentin.sturm@lcm.at
veronika.putz@lcm.at
julia.mayer@lcm.at

ABSTRACT

Cutting tool wear needs to be monitored closely to ensure good quality of machined parts. However, manual inspection is both expensive and time consuming, therefore there is a need for automated monitoring methods. We present a technique that can reconstruct the cutting tool surface in 3D, allowing a spatial estimation of the tool wear with high accuracy. The reconstruction allows an automated direct monitoring method that estimates at any time the cutting tool condition, avoiding conversion work and major quality issues. The optical measurement setup consists of a hardware triggered line scan camera that registers the spinning cutting tool's shadow inflicted by a collimated backlight. We show how to leverage the 1D line scan signal acquired at varying cutting heights of the tool into a full 3D reconstruction. The progression of tool wear may thus be monitored by comparing the reconstructed shape to previous measurements. To this end we show a methodology for tool wear quantification. Additionally, to assess the measurement technique, an accuracy analysis with ground truth geometry was performed. The technique was applied to multiple degrading drilling tools. By automation of the cutting tool health monitoring, retrofitting this technology on a conventional machining center would transform it into an Industry 4.0 compatible (smart) machining center utilizing off-the-shelf optical equipment with moderate costs.

Rob Salaets et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

The wear of cutting tools is unavoidable in any cutting process, given the extreme levels of friction, stress and temperatures subjected to the tool. The relation of tool wear to part quality is evident: the dimensional accuracy and surface finish of the work piece are directly affected by a worn cutting tool. Furthermore, the degradation of cutting tools can occur unpredictably. This is the case when cutting composites, due to the composite laminates characteristics, such as inhomogeneity and anisotropy and to the layered structure and the extreme abrasiveness of fiber reinforcements (Bey-Temsamani, Ooijevaar, & Depraetere, 2019). Accordingly, there is a need for close monitoring of cutting tool wear, as a requirement for good quality of machined parts.

In this paper we investigate the feasibility of a 3D reconstruction of a rotating geometry using a single line scan camera for metrology purposes. We apply this with a particular focus on CNC cutting tools. Building on our previous research presented in (Bey-Temsamani et al., 2019), we propose a direct optical monitoring technique that can reliably detect tool wear based on periodic automated measurements. To this end, we use a line scan camera and collimated light to reconstruct the tool's geometry out of shadow scans of a spinning tool.

Cutting tool monitoring has been researched before, and is generally classified either direct or indirect monitoring. Direct methods try to assess measurements of the tool wear itself and indirect methods deduce the tool wear from sensor signals on the cutting machine implicitly.

In the field of indirect monitoring, multiple approaches exist. In (Krishnakumar, Rameshkumar, & Ramachandran, 2018) a feature based machine learning approach is proposed, where

multiple type of emissions, namely acoustic and vibration data are processed. The authors report high a classification efficiency. The authors of (Zhu & Zhang, 2019) propose a generic wear model which allows for prediction of milling force needed and remaining tool life.

A review on indirect methods can be found in (Kuntoğlu et al., 2020), where multiple articles are presented concerning monitoring schemes using vibration, heat and other emissions. Direct methods enable us to quantify the deviations or wear in a direct way. This can provide more insights than indirect monitoring, at the price of higher hardware costs. The wear can be spatially localized, allowing a classification of degradation processes. Furthermore, indirect methods are to be made robust to different cutting parameters, work piece materials and tool geometries, while direct methods do not have this requirement. In (Bagga, Makhesana, Patel, & Patel, 2021), the authors propose a direct method, namely a set-up with a camera and an image processing step to determine maximum flank wear measurement on CNC lathe machine. The authors of (Peng, Pang, Jiang, & Hu, 2020) use a similar setup. The resulting grayscale images are processed using software which allows the user to quantify tool wear. Both methods use machine vision techniques to analyse 2D images, but do not attempt a geometric reconstruction. The approach in the work of (Čerče, Pušavec, & Kopač, 2015) is most similar to ours, they use a 2D laser displacement sensor to measure the cutting tool surface. The approach also requires a motorized linear translation stage to displace the sensor over the static tool. In contrast, our approach leverages the mobility of the CNC machine tool and achieves similar repeatability with cheaper, more primitive camera technology.

In the industrial state of practice, the cutting tools are replaced preventively based on a visual inspection. This method has many limitations that makes it inefficient as a current practice. First, operators need to check the status of the cutting tool regularly by stopping the machine every now and then (Bey-Temsamani et al., 2019). A naked eye visual inspection can only be up to 100µm accurate, higher accuracies can be attained by taking the tool to a dedicated microscope. However, this is a lengthy and costly process.

Our proposed method allows the automation of cutting tool quality monitoring, because the measurement setup is in the machine tool working volume. A short (~ 30 seconds) measurement cycle can be executed after some CNC operations have been done, without stopping the machine. A comparison of measurements of a tool in pristine condition and after use, will reveal the amount of tool wear that has occurred. The simplicity of the optical setup allows for retrofitting existing CNC machining centres with ease.

The following section describes the optical setup and the necessary data acquisition modalities. Section 3 introduces the processing steps of the 3D reconstruction method. In Section

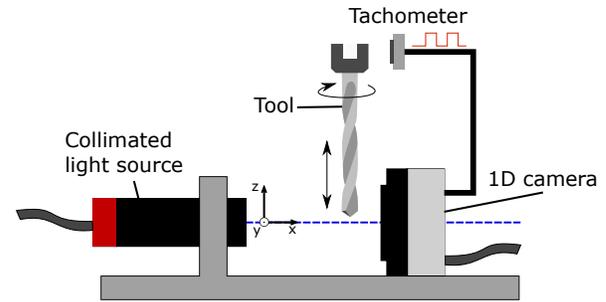


Figure 1. An schematic overview of the optical setup.

4 we analyse the accuracy and repeatability of our method. Section 5 presents the application of our method on a dataset of worn drilling tools. A concluding remark is made in Section 6.

2. OPTICAL SETUP AND DATA ACQUISITION

The hardware of the measurement setup consists of a line scan camera, a collimated light source and an optical tachometer. Figure 1 illustrates their arrangement. During measurement, the tool remains clamped in the spindle. The CNC machine places the tool between camera and light source, increments its vertical position in small steps and rotates the tool at the lowest rotational speed (≈ 3500 rpm). The cutting tool is backlit using a collimated light source, which is placed at a distance of 140mm from the tool and 150 mm from the front face of the camera. To measure the actual rotation speed and to align the captured data with the angular orientation of the tool, an optical tachometer is used. It captures a white marking on a dark background fixed onto the clamping nut and outputs a triggering signal (a rectangular wave with a single rising edge at each revolution of the tool). The tachometer signal is used to trigger a monochromatic line scan camera (Basler Racer type raL2048-48gm, pixel size $7\mu\text{m}$, 2046 pixel/scan), which subsequently captures 2500 scan lines at a repetition rate of 50kHz at each rising edge. One frame is stored at each depth level together with the CNC coordinates from the controller. The camera speed and pixel size need to be chosen to match the requirements imposed by the minimal rotational speed of the spindle and the tool size respectively.

3. TOOL RECONSTRUCTION METHOD

3.1. Processing pipeline

The geometry reconstruction consists of multiple stages, they are depicted in Figure 2. For each vertical depth step a shadow trace of the spinning tool (line scan image) is recorded. Using a thresholding approach the shadow edge locations on the left and right side are accurately estimated, as described in Subsection 3.2. These signals contain some duplicated information, they can be aggregated to improve the estimates. Subsequently, the cross section of the tool can be reconstructed.

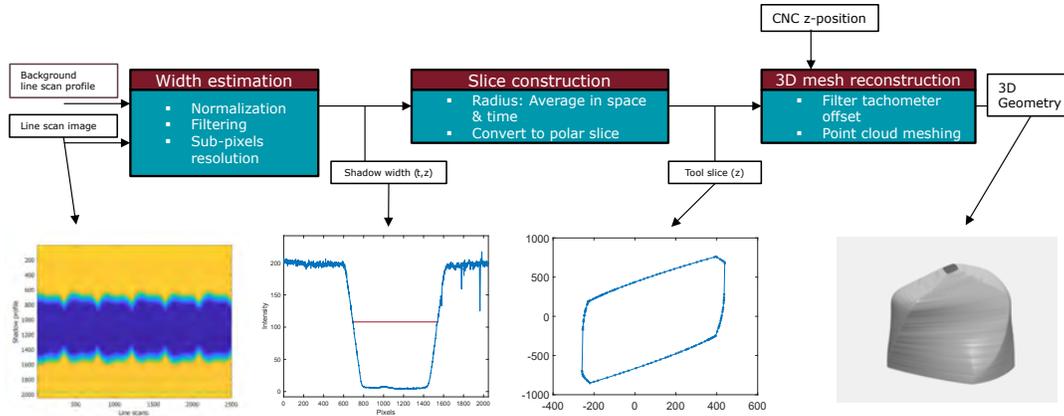


Figure 2. Overview of the 3D reconstruction pipeline.

These aspects are presented in Subsections 3.3 and 3.4. A point cloud can be obtained by stacking up the reconstructed cross sections. A straightforward meshing algorithm is introduced in Subsection 3.5.

3.2. Shadow width estimation

Figure 3, top left, shows raw data recorded from a helix-shaped cutting tool¹ at a fixed vertical position of the CNC machine as 2D image. The vertical axis displays intensity values recorded by the camera in a single scan as gray value. The horizontal axis shows all 2500 scanlines recorded from the rotating tool in a single experiment, i.e. data recorded while the tool performed three full rotations about 360°. Although a collimated light source was used, the transition from areas with low intensity (in which the light source was shadowed by the tool) and areas with high intensity (in which the sensor was directly illuminated by the light source) is soft, it covers a width of 0.63 mm. The soft shadow needs to be sampled sufficiently for accurate estimates, which requires a line scan camera with pixel size two orders of magnitude smaller (See Section 2). Due to the soft shadow effect, the relation between the width of the shadow created by the tool and the actual projected width is not straightforward. Especially in data recorded from the tip of the tool (which is highly relevant for measuring tool wear), the transition between illuminated and shadowed areas strongly influences the measurement. In this subsection, we present two thresholding techniques.

Besides the soft transition between bright areas and shadowed areas, as visible in Fig. 3, top left, also some sensitivity variations of the camera appear as horizontal lines. To account for these disturbances in an raw image I , the image was smoothed by applying a simple 2D-moving average filter

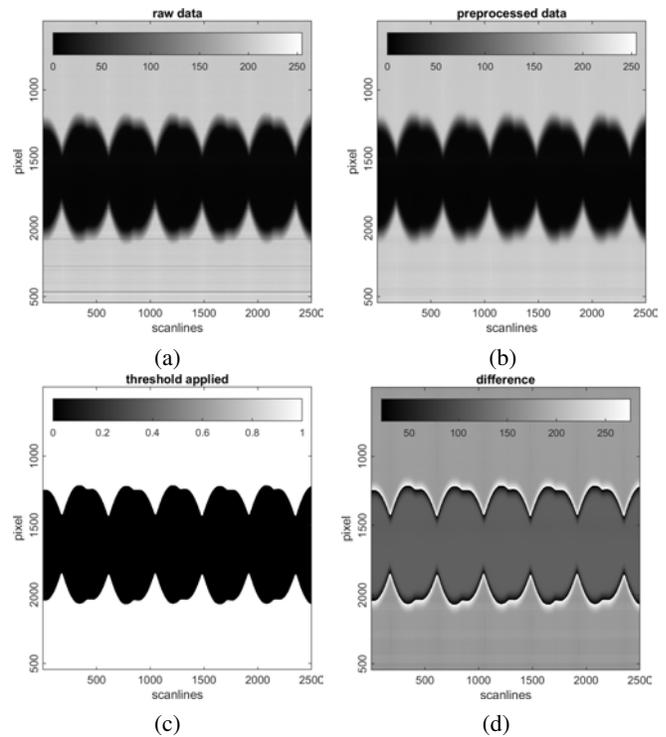


Figure 3. Line scan processing pipeline: The recorded raw data (a) is normalized, and sensitivity deviations of the used camera are compensated (b). A binary threshold is applied (c), and the deviation of the preprocessed data from the binarized image is further evaluated (d)

¹Izartool 6,00 mm-DRILL BIT HSSE DIN338N

of length 61:

$$\hat{I}_{i,j} = \frac{1}{(\hat{k}_1 - \hat{k}_2 + 1)(\hat{l}_1 - \hat{l}_2 + 1)} \sum_{k_1=\hat{k}_{1,1}}^{\hat{k}_{1,2}} \sum_{k_2=\hat{k}_{1,1}}^{\hat{k}_{1,2}} I_{i+k_1, j+k_2},$$

with

$$\hat{k}_1(i) := \max\{-i + 1, 30\}, \quad \hat{k}_2(i) := \min\{n - i, 30\},$$

$$\hat{l}_1(j) := \max\{-j + 1, 30\}, \quad \hat{l}_2(j) := \min\{n - j, 30\}.$$

Figure 3, top right, shows our exemplary data after such a pre-processing step. Line scan images without the tool in front of the camera can serve as a background image which can be subtracted from the line scans with a tool present. This method works well when the noise is dominated by variations in pixel sensitivity.

Adaptive threshold The first thresholding method, denoted with M_a , rescales each scan line individually to contain values between 0 and 1, and applies a transformation afterwards of the form:

$$T(\hat{I}_{i,j}, th) := \begin{cases} 1 & \text{if } \hat{I}_{i,j} \geq th, \\ 0 & \text{if } \hat{I}_{i,j} < th. \end{cases}$$

The parameter th is determined for each scan line by minimizing the following error term

$$th(j) := \min \left\{ th_2, \underset{th \in R}{\operatorname{argmin}} \sum_{i=1}^{2046} |T(\hat{I}_{i,j}, th) - \hat{I}_{i,j}| \right\},$$

where th_2 is introduced to reduce diameter- overestimation for small shadow widths. It was learned on a different dataset with a different drill, such that this algorithm minimized the overall reconstruction error with respect to mean absolute deviation. Figure 3, bottom left, shows the pre-processed data after applying a threshold using the function T . The deviation from this threshold is illustrated in Fig. 3, bottom right. After minimizing with respect to each scan line, we are able to extract the left and right shadow edge positions by taking the minimal and maximal position where the value is 1. These values are multiplied with the pixel size to return an estimate in μm .

Fixed threshold The fixed threshold method, denoted with M_f , compares normalized line scans to a single threshold intensity. It was empirically found that a threshold of halfway ($th = 0.5$) between the darkest shadow intensity level and background intensity level is most robust to the influence of the tool position. The distance between the cutting tool and camera widens the shadow edge, as no light source is perfectly collimated. The shadow edges are now found by searching for the zero crossing points of $\hat{I}_{i,j} - 0.5$. This done by

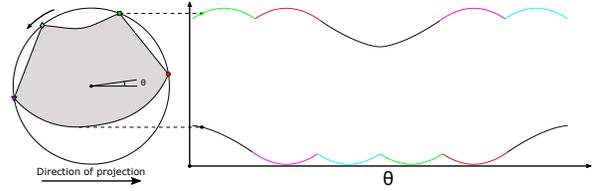


Figure 4. Illustration of shape projection and duplicated information in shadow edges.

calculating the analytical roots of a least-squares parabolic fit through the points around the indices where $\hat{I}_{i,j} - 0.5$ changes sign. Note that the threshold was optimized for robustness, instead of the actual projection width of the tool. Therefore, the final shape reconstruction needs to be scaled by a scaling factor that minimizes the reconstruction error.

3.3. Filtering in time, space and phase

For each depth level, the shadow edges are registered during multiple rotations of the tool. Due to the collimated light, the shadow of the tool is an orthogonal projection on the line sensor of a planar cross section of the geometry, for the current depth level (See Figure 4).

The estimated shadow edge locations correspond to the extrema of the projected shape over a period of time. Both left and right edge locations ($s_l[t]$, $s_r[t]$) contain information of projected shape. This is depicted in Fig 4, it is clear that both the top and bottom traces of the shadow edge capture the same periodic pattern but inverted and phase shifted by a half revolution. For robustness against sensor noise the signals can be averaged over I periods with length T :

$$s_{avg}[t] = \frac{1}{I} \sum_{i=0}^{I-1} s[t + iT]$$

Afterwards a phase shifted average over left and right combines the information in both shadow edges. The result is a filtered half projection width:

$$h[t] = \frac{1}{2} (s_{r,avg}[t] - s_{l,avg}[\operatorname{circ}(t - T/2, T)])$$

Here circ wraps around negative indices to the end of the array for a periodic extension. We estimate the period length T by finding the second local maximum of the cross-correlation function of s_r or s_l with a smaller subsection. A rough estimate of the spindle's angular velocity helps to identify the correct peak. When a speed controlled spindle is used, the angular velocity of tool can be regarded as constant, therefore the independent variable t can be regarded as the rotation angle, as shown in Figure 4.

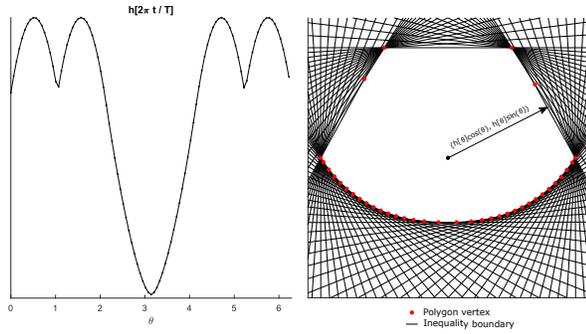


Figure 5. Illustration of duplicated information in shadow edges.

3.4. Slice reconstruction

The inverse problem at hand is similar to computed tomography. However, in this case we consider an opaque geometry instead of translucent imaging. Computed tomography methods result in a discretization of the two-dimensional space. Recognizing these important differences, we present an approach where the shape reconstruction is obtained by solving the vertex enumeration problem with a primal-dual polytope method (Bremner, Fukuda, & Marzetta, 1998). The vertex enumeration problem computes the vertices of the n-polytope that satisfies a set of inequalities. Figure 5 displays how the resulting convex polygon is constructed. The system of inequalities is given by:

$$\mathbf{Ax} \leq \mathbf{b}$$

With the rows of \mathbf{A} and elements of \mathbf{b} given by:

$$a_t = (\cos(2\pi t/T) \quad \sin(2\pi t/T)),$$

$$b_t = (h[t]), t = 0, \dots, T - 1$$

See (Bremner et al., 1998) for a detailed description of the primal-dual method for vertex enumeration. By projecting the shape onto the line sensor, information is lost. Indeed, only the convex hull of the shape can be reconstructed. However, convexity is only enforced in the two-dimensional cross section perpendicular to the axis of rotation. Vertical concavity can be present in the entire three-dimensional reconstruction. Depending on the application, that is an important limitation of the technique. We argue that for tool wear detection it is not a critical limitation, because tool wear occurs primarily at the convex ridges of the geometry (e.g. cutting edge and drill margin for drilling tools).

The acquisition of line scan frames is synchronised by a digital tachometer signal. However, we have observed a phase uncertainty with sample standard deviation $\sigma \approx 1.25^\circ$. This error would be a dominant factor of the total inaccuracy, but it can be reduced by filtering over all cross sections. Neighbouring cross sections have a similar shape and therefore the phase difference can be estimated with a circular cross correlation.

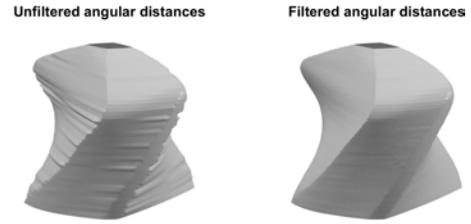


Figure 6. Illustration of the effect of filtering the angular distances.

The array of angular differences can be filtered to smooth out the noise. Filtering should not reduce the total angular twist of the tool, therefore a filtering technique should be applied with care. We employ a one-dimensional Laplacian smoothing (Lo, 1985). Let $d[i]$ be the angular differences between two cross sections and $s[i] = \sum_{k=0}^i d[k]$ for $i = 0, \dots, I$. The smoothing updates are given by:

$$s' = s - \lambda \mathbf{L}s$$

$$\mathbf{L} = \begin{pmatrix} -0.5 & 1 & -0.5 & & \\ & \ddots & \ddots & \ddots & \\ & & -0.5 & 1 & -0.5 \end{pmatrix}$$

where updates are applied multiple times on the inner points: $i = 1, \dots, I - 1$. We found that 20 iterations with $\lambda = 0.8$ works well. The effect of this filtering is apparent in Figure 6.

3.5. Point cloud and mesh construction

Solving the vertex enumeration problem for each depth level results in a point cloud that can be meshed into a triangular surface. Because the points are effectively coming from a set of planer cross sections of the tool, this structure can be exploited for a simplified meshing method. The problem reduces to connecting the points of two subsequent cross sections into triangles. A good mesh has minimal skewness of the triangles. To this end, we employ the Dynamic Time Warp (DTW) transform on the points in cylindrical coordinates. The DTW finds the closest match between the lists of angular coordinates, the output consists of two monotonically increasing arrays of indices that contain a many-to-many mapping between indices. Multiplicity of a node results in duplicated indices in the list.

Listing 1 describes the meshing algorithm. Note that the point cloud is not altered, the output is merely a selection of triplets for a surface triangularization. The process is further illustrated in Figure 7, green triangles are created out of the black pairs that are outputted by the DTW transform.

Data: $(r_i[k], \theta_i[k], z_i[k])$ cylindrical coordinates of the points of at every depth level $i = 1, \dots, I$

Result: T the set of triangles

$T \leftarrow \emptyset$

for $i=2$ **to** I **do**

$ia, ib \leftarrow \text{DynamicTimeWarp}(\theta_{i-1}[k], \theta_i[k])$

for every pair (a_k, b_k) **in** (ia, ib) **do**

if $a_k == a_{k-1}$ **then**

 Append($T, \text{GlobalIndex}(a_k, b_{k-1}, b_k)$)

else if $b_k == b_{k-1}$ **then**

 Append($T, \text{GlobalIndex}(a_{k-1}, b_k, a_k)$)

else

 Append($T, \text{GlobalIndex}(a_{k-1}, b_{k-1}, b_k)$)

 Append($T, \text{GlobalIndex}(a_{k-1}, b_k, a_k)$)

end

end

end

Algorithm 1: Triangle meshing algorithm. The *GlobalIndex* function maps three local indices to the global indices to be used in a connectivity matrix.

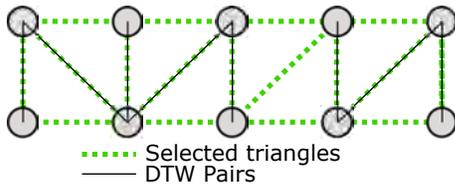


Figure 7. Illustration of the DTW meshing method.

4. ACCURACY ANALYSIS

4.1. Accuracy analysis of width estimation

In this section we present an accuracy analysis for our fixed threshold and adaptive threshold shadow width estimation methods, which were defined in Section 3. We recorded three measurements of the same tool, and in the following we compare the reconstructed shapes based on these three measurements with the ground truth. Afterwards, we discuss the repeatability by comparing our three different reconstructions with each other.

A ground truth geometry of the drill used for this analysis was obtained with a 3D scanner (GOM Atos Core 200). For structured light scanning of metal cutting tools an anti-reflective coating spray is needed. Considering these conditions an accuracy of 15-20 μm can be expected. This puts a lower bound on our accuracy analysis.

For our first analysis, we considered a cross section at each z-level of our ground truth and compared it with the respective reconstruction. To this end, we randomly sampled points from the outlines and subtracted the estimated radius from the ground truth. The results thereof are depicted in two histograms in Figure 8 and Figure 9. We provide some performance measures, namely Mean Signed Difference (MSD), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and 5%- and 95%-quantiles in Table 1.

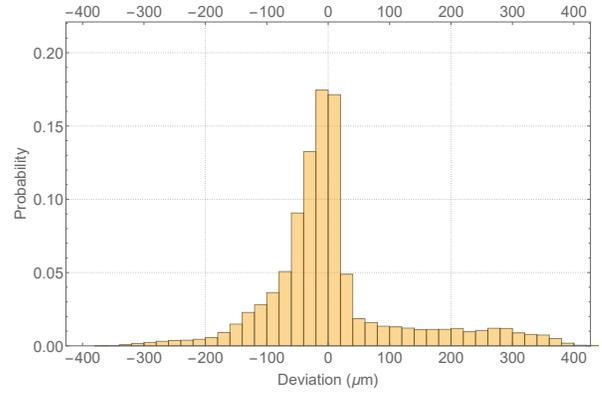


Figure 8. Histogram of deviation values in μm when reconstructing the shape with the adaptive threshold method M_a .

In Table 1 we can see that the first two average error met-

Table 1. Performance measures in μm for two considered approaches with respect to shape reconstruction

Approach	MSD	MAE	RMSE	$q_{0.05}$	$q_{0.95}$
M_a	7.85	72.70	111.98	-133.66	276.74
M_f	-12.40	73.71	107.68	-160.81	222.05

rics and the 5%-quantile values are more favorable for the M_a Method, while the 95%-quantile and the RMSE is better in case of applying the M_f reconstruction. In general, the results do not indicate a large difference in reconstruction quality.

For our second analysis, we analysed our three reconstructions by examining all three possible pairwise comparisons and averaged the results. Similar to our analysis before we also calculated some performance metrics in Table 2 below. When comparing Table 1 and 2 we can observe that the re-

Table 2. Performance Measures in μm for our two considered approaches with respect to repeatability.

Approach	MSD	MAE	RMSE	$q_{0.05}$	$q_{0.95}$
M_a	-0.507	5.697	8.163	-14.0	12.78
M_f	-0.741	4.416	6.291	-11.85	9.67

peatability error is one order of magnitude smaller than the reconstruction error for each respective metric. Moreover, it shows that with respect to repeatability the M_f Method performs superior, with better values than M_a for each performance measure except MSD. The resulting error distributions are depicted in Figure 10 and Figure 11.

4.2. Geometry alignment challenges

Some information is lost during the reconstruction, i.e. the convex hull of a cross section can be reconstructed (Subsection 3.4). This makes the comparison with a ground truth

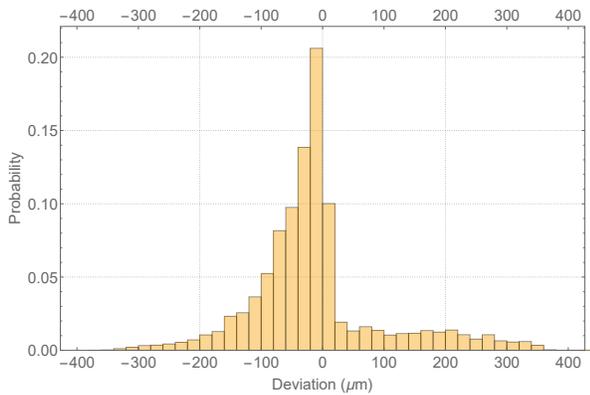


Figure 9. Histogram of deviation values in μm when reconstructing the shape with the fixed threshold method M_a .

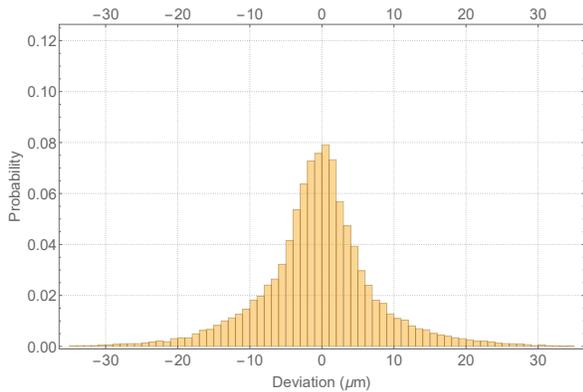


Figure 10. Repeatability error of reconstructions using method M_a .

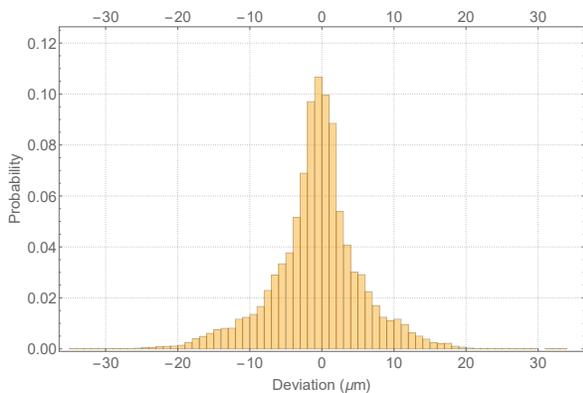


Figure 11. Repeatability error of reconstructions using method M_f .

point cloud challenging. A prerequisite for a valid comparison is a good alignment of the geometry. Rigid point-set registration is typically used for such alignment, we have used the Iterative Closest Point (ICP) algorithm (Chen & Medioni, 1992). Our experiments have proven that in order to have a well converging alignment, the ground truth point cloud needs to be processed to exclude the points that would be lost in reconstruction. When aligning two measurements produced by our method, this problem does not occur.

5. APPLICATION: TOOL WEAR MONITORING

The geometry reconstruction method presented in Section 3 provides the foundation for a cutting tool wear detection method. By comparing the geometry of a measurement at the beginning of the cutting tool’s life with measurements after use, the tool wear can be spatially visualized and quantified. This kind of analysis enables an expert to assess the severity of the tool wear, and take appropriate actions. A simple automated replacement method can be obtained by comparing the quantified tool wear to a threshold.

For alignment of two measurements the ICP algorithm is used, as presented in Section 4. For this purpose the entire point clouds of both measurements can be used. After alignment of the geometry, cutting tool wear can be quantified by a Euclidean distance metric. As a tool wear metric, we use point-to-closest-triangle distance between the triangular surface of the original measurement and the point cloud of the worn cutting tool measurement. These per-vertex values can be put in a histogram and compared to the point-to-closest-triangle repeatability histogram. This is a fair comparison if the vertices are distributed uniformly over the surfaces. However, the geometry coming out of our reconstruction does not exhibit this property. Indeed, there are more points in the curved areas, because there are more line scans associated with these areas. To account for this imbalance, weighted histograms can be of use. Each vertex is weighted by its local area:

$$w_i = \sum_{j \in T_i} A_j / 3$$

where, T_i are the neighbouring triangles of vertex i and A_j is the triangle area of triangle j . A weighted histogram then aggregates weight values by sum in each bin, instead of by count. Tool wear can now be assessed using an area weighted histogram of the point-to-closest-triangle distance. See Figure 12 (Appendix) for an illustration of such weighted histogram.

To validate our method with respect to tool wear detection, we have done accelerated degradation tests on drilling tools. In order to obtain wear quickly, a hard work piece material (stainless steel) and wrong cutting parameters were used on low cost tools. A quantitative and qualitative overview is given in Figure 13 (Appendix): the first row visualizes

the tool wear metric localized on the worn tool reconstruction, the point cloud of the original geometry is visualized in black; the second row displays the worn geometry with specular shading enabled in order to give a qualitative comparison of the cutting edge. Therefore the bottom row is rotated 90 degrees counter-clockwise. The tool wear metric is also presented in weighted histogram form in Figure 12. For illustrative purposes the degradation is driven to an unusable level. However, it is clear how the progression of wear can be seen in all three visualization forms. In Figure 13, most notably the corner point of the flank face has worn off after three holes and the chisel edge becomes smaller after each hole. The bottom row reveals how the initially sharp cutting edge has been chamfered and is blunt after two holes. In Figure 12 this progression of tool wear is also reflected. The first weighted histogram illustrates the repeatability in terms of point-to-triangle-distance. Subsequent histograms show a widening that corresponds to tool wear progression. Negative values are most indicative for tool wear. Positive values are present because of imperfect alignment of the non matching geometries. A simple quantification of the total tool wear considers the sum of the weights of the bins with magnitude above a certain threshold. This value can be used to trigger maintenance actions. This experiment validates our approach towards tool wear analysis using the proposed reconstruction method. It proves feasibility of the reconstruction method for tool wear monitoring, with moderate costs and off-the-shelf components.

Further efforts can investigate robust estimation methods in contaminated environments: cutting oil, coolant, chips, milling waste. Figure 12 also motivates investigation of the use of maximum curvature as a tool wear parameter.

6. CONCLUSION

In this article we presented a method for surface reconstruction of rotating geometry. By leveraging shadow traces of a spinning tool, the 2D convex hull of the geometry can be reconstructed. The accuracy of the proposed method proves sufficient for detecting tool wear. We compared two methods for shadow width estimation and see comparable performances when applying our adaptive threshold method M_a or our fixed threshold method M_f with respect to reconstruction quality, while the repeatability shows less variance when using the fixed threshold method M_f . When considering the Mean Absolute Errors in Table 1 with values around $70 \mu m$ we attain values better than the accuracy possible by human visual inspection and the industry standard of $100 \mu m$. The results from our repeatability analysis with mean errors as low as around $10 \mu m$ and the tool wear experiment results from Section 5 corroborate the feasibility of our approach for tool wear monitoring. The difference in distance distribution due to increasing wear is easily visible and traceable. Moreover, due to the low-cost set-up with autarkic equip-

ment, retro-fitting represents a viable way of transforming a conventional machining center into a smart one.

ACKNOWLEDGEMENT

This research was supported by Flanders Make, the strategic research centre for the manufacturing industry. This work has also been supported by the COMET-K2 Center of the Linz Center of Mechatronics (LCM) funded by the Austrian federal government and the federal state of Upper Austria. The authors would like to thank Stijn Helsen for his efforts on the optical setup and data acquisition.

REFERENCES

- Bagga, P., Makhesana, M., Patel, K., & Patel, K. (2021). Tool wear monitoring in turning using image processing techniques. *Materials Today: Proceedings*, 44, 771–775.
- Bey-Temsamani, A., Ooijejaar, T., & Depraetere, B. (2019). An assessment of two technologies for high performance composite machining; adaptive fixturing and in process tool profile monitoring. *Procedia CIRP*, 85, 201-206. Retrieved from <https://www.sciencedirect.com/science/article/pii/S2212827119312831> (2nd CIRP Conference on Composite Material Parts Manufacturing, 10-11 October 2019, Advanced Manufacturing Research Centre, UK) doi: <https://doi.org/10.1016/j.procir.2019.09.023>
- Bremner, D., Fukuda, K., & Marzetta, A. (1998, Oct 01). Primal—dual methods for vertex and facet enumeration. *Discrete & Computational Geometry*, 20(3), 333-357. Retrieved from <https://doi.org/10.1007/PL00009389> doi: 10.1007/PL00009389
- Chen, Y., & Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3), 145-155. Retrieved from <https://www.sciencedirect.com/science/article/pii/026288569290066C> (Range Image Understanding) doi: [https://doi.org/10.1016/0262-8856\(92\)90066-C](https://doi.org/10.1016/0262-8856(92)90066-C)
- Krishnakumar, P., Rameshkumar, K., & Ramachandran, K. (2018). Feature level fusion of vibration and acoustic emission signals in tool condition monitoring using machine learning classifiers. *International Journal of Prognostics and Health Management*, 9(1).
- Kuntoğlu, M., Aslan, A., Pimenov, D. Y., Usca, Ü. A., Salur, E., Gupta, M. K., ... Sharma, S. (2020). A review of indirect tool condition monitoring systems and decision-making methods in turning: Critical analysis and trends. *Sensors*, 21(1), 108.
- Lo, S. H. (1985). A new mesh generation scheme for arbitrary planar domains. *International Journal for Numerical*

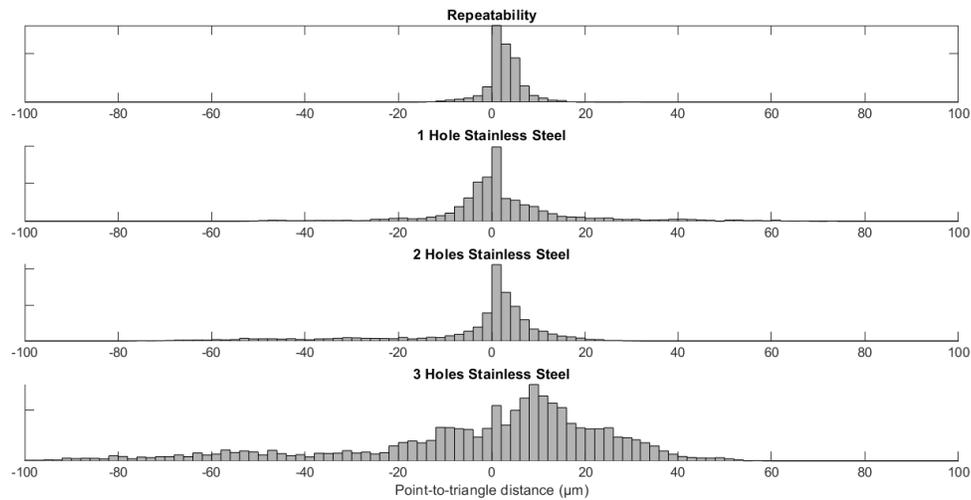


Figure 12. Progression of the weighted point-to-triangle distance histograms of a worn drill.

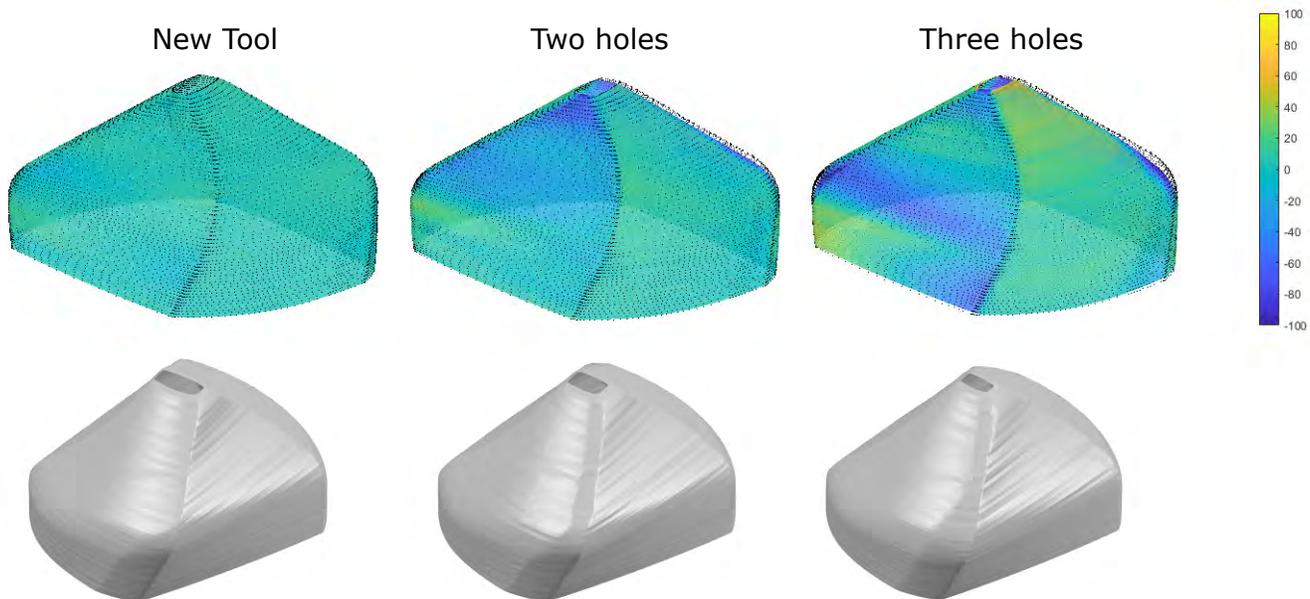


Figure 13. Visualized tool wear metric and shaded reconstructions of a (accelerated) degrading drilling tool. Left: unused drill; Middle: two holes in stainless steel; Right: three holes in stainless steel.

Methods in Engineering, 21(8), 1403-1426. Retrieved from <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1620210805> doi: <https://doi.org/10.1002/nme.1620210805>

Peng, R., Pang, H., Jiang, H., & Hu, Y. (2020). Study of tool wear monitoring using machine vision. *Automatic Control and Computer Sciences*, 54(3), 259–270.

Zhu, K., & Zhang, Y. (2019). A generic tool wear model and its application to force modeling and wear monitoring in high speed milling. *Mechanical Systems and Signal*

Processing, 115, 147–161.

Čerče, L., Pušavec, F., & Kopač, J. (2015). A new approach to spatial tool wear analysis and monitoring. *Strojniški vestnik - Journal of Mechanical Engineering*, 61(9), 489-497. Retrieved from <https://www.sv-jme.eu/article/a-new-approach-to-spatial-tool-wear-analysis-and-monitoring/> doi: 10.5545/sv-jme.2015.2512

Data-Driven Fault Detection for Transmitter in Logging-While-Drilling Tool

Karolina Sobczak-Oramus¹, Ahmed Mosallam², Caner Basci³, and Jinlong Kang⁴

¹ Schlumberger, Nowogrodzka Street 68, Warsaw, Mazowieckie, 02-014 Poland
KSobczak@slb.com

^{2,4} Schlumberger, 1 Rue Henri Becquerel, 92140 Clamart, France
AMosallam@slb.com
JKang5@slb.com

³ Schlumberger, 2-2-1 Fuchinobe, Chuo Ward, Sagami-hara, Kanagawa 252-0206, Japan
CBasci@slb.com

ABSTRACT

Logging tools widely used in the oil and gas industry are exposed to demanding environmental conditions that can lead to faster degradation and unexpected failures. These events can reduce productivity, delay deliverables, or even bring entire drilling operations to an end. However, such accidents can be avoided using a prognostics and health management approach. This paper presents a data-driven fault detection method for transmitter in logging-while-drilling tool adopting a support vector machine classifier. The health analyzer determines the component's physical condition in just a few minutes, demonstrating an exceptional value for both field and maintenance engineers. This work is part of a long-term project aimed at constructing a digital fleet management system for downhole testing tools.

1. INTRODUCTION

Prognostics and Health Management (PHM) combines the knowledge and experience from several disciplines such as engineering science, computer science, reliability engineering, and more, to assess a product's degradation and reliability. PHM has emerged recently as a momentous technology that makes an impact on maintenance practices for different industrial systems (Vachtsevanos, Lewis, Roemer, Hess, & Wu, 2006). Design, monitoring, and maintenance of complex systems such as aircraft, manufacturing, and industrial processes, and more have undergone a real transformation, being more data-driven thanks to the usage of PHM practices. PHM technologies are quickly evolving, the customer

Karolina Sobczak-Oramus et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

base for these technologies is expanding, and their potential application domains are increasing at a phenomenal rate (Vachtsevanos et al., 2006).

Prognostics and health management consist of the following main pillars: fault detection, fault diagnostics, fault prognostics, and decision support (Mosallam, Medjaher, & Zerhouni, 2016). Fault detection is used to determine that a problem has occurred within the monitored component. Fault diagnostics is the process of identifying faults and their causes. Fault prognostics helps estimate the remaining time left for a system or a component before it fails. Finally, decision support is used to select the proper maintenance actions based on the information gathered about the monitored system status.

SonicScope multipole sonic-while-drilling service tool (Figure 1) is a multi-function logging-while-drilling (LWD) tool developed for oil well drilling applications available in the following collar sizes:

- $4\frac{3}{4}$ in,
- $6\frac{3}{4}$ in,
- $8\frac{1}{4}$ in,
- 9 in.

It is commonly used in conjunction with other LWD equipment during the drilling phase of well construction.



Figure 1. SonicScope service tool in different collar sizes

LWD tools are often exposed to demanding environmental conditions such as shocks, vibrations, pressures, and elevated temperatures (Kirschbaum et al., 2020). Consequently, the degradation rate of the subsystems in the tools can increase over time resulting in tool failures. Hence, the information gained from the tools can be inaccurate, and this could compromise the operation. As a result, the deliverables can be delayed until the tool is fixed or, in a worst-case scenario, the entire operation may be canceled. Such situations lead to non-productive time and financial losses. To avoid such failures, after each run, field engineers are required to check the tool condition. Each tool consists of multiple different subsystems that contain many parts. To assess the overall tool condition, field engineers should analyze sensor signals for each part recorded during tool operation. Consequently, they need to decide if the tool and its subsystems are healthy and can be used again in the next run. Depending on the subsystem, they must decide if the tools should be repaired or junked. However, due to the large number of data channels generated at a record rate, which results in millions of data points from a single run, developing manually a solid analysis is extremely challenging (Mosallam, Laval, Youssef, Fulton, & Viassolo, 2018). A manual analysis of this data is time-consuming in an environment where time is critical, and the complexity of the signals limits the effectiveness of manual analysis.

Alternatively, the critical subsystems in the tool can be identified and a domain expert can select the channels that contain information about the tool condition and possible degradation of each subsystem. Statistical features that indicate the degradation of the system in time are extracted from the selected channels. These features can be used to build machine learning models that estimate the tool condition. Using the SonicScope service tool, a transmitter subsystem was identified as one of the most critical components by failure modes, effects, and criticality analysis (O'Connor & Kleyner, 2012). Thus, a fault detection algorithm was developed to help the field engineers identify whether the component behaved as expected or not.

In this paper, we present a data-driven fault detection method for transmitters in drilling tools. The transmitters are of distinct types and sizes depending on the version of the tool. As a result, the models and features extracted from the raw data differ between the types and sizes of the transmitters. The method is based on extracting relevant features that can identify healthy and faulty transmitters. A support vector machine model is trained on the features extracted from different runs labeled as healthy or faulty by a domain expert.

This paper is structured as follows. A literature review is presented in Section 2. Section 3 presents a description of the transmitter subsystem. The method and the results for the fault detection model are presented in Section 4. Finally, Section 5 concludes the paper.

2. RELATED WORK

Transmitters are widely used in different types of equipment in different industries. Thus there are plenty of research works on transmitter fault detection. For instance, (Ganesh Kumar, Insozhan, & Parthasarathy, 2019) uses a fuzzy inference system to monitor transmitter circuit conditions in a wireless sensor network. (Tugova, Salov, & Bushuev, 2021) presented a fault diagnosis method of a pressure transmitter based on output signal noise characteristics. (S. Liu, Xu, Li, Zhao, & Li, 2018) proposed a hybrid fault diagnosis model for transmitters in water quality monitoring devices based on multiclass support vector machines in combination with rule-based decision trees. (C. Liu, Chen, Zhang, & Wang, 2018) introduced a fault diagnosis application of a short wave transmitter based on a stacked auto-encoder.

Generally, transmitters for different devices have different designs, which makes the fault detection method of the transmitter cannot be used universally. The device studied in this paper is a specific tool used for logging when drilling oil and gas wells. To the best of our knowledge, there is no published research work about LWD transmitter fault detection or fault diagnosis. Therefore, our study can give an idea of how to proceed with the specific case of transmitter's diagnostics.

3. TRANSMITTER

SonicScope service combines high-quality monopole and quadrupole measurements to obtain compressional, shear, and Stoneley data in all formations and across a wide range of hole sizes from surface to true depth. Combined with a full-characterized tool design, simplified and automated operations, and advanced processing techniques, the SonicScope service delivers robust, accurate, and reliable acoustics measurements for many applications from petrophysics to cement evaluation.

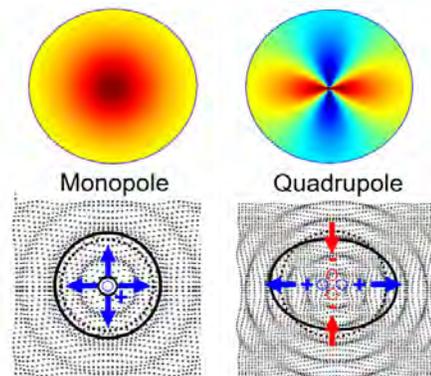


Figure 2. Monopole and quadrupole transmitters

SonicScope service carries two types of transmitters (Figure 2): monopole and quadrupole. Monopole transmitter is mainly used to measure the formation slowness for fast formations and quadrupole is used to measure the formation

slowness for slow formations. Acoustic waves generated by the transmitters are captured by the array of receivers with a total quantity of 48. These receivers are located around all four sides of the tool. The transmitter consists of piezoceramic elements that generate acoustic waves when excited with high voltage. As a tool diagnostics, the tool records each transmitter’s excitation voltage to interpret the health of the transmitter.

4. PROPOSED METHOD

We present a data-driven fault detection method to analyze the health of a transmitter. There are a couple of steps to conduct the data-driven PHM projects. The first phase of the fault detection approach is to conduct the data inventory, which allows collecting the data from different sources and services. The second step is to identify the scope, particularly perform an analysis of different failure modes and select the most critical ones. Several different problems can cause transmitter failures, such as low insulation resistance, different physical damages, and more. Subject matter experts (SMEs) select the critical problem in terms of priority, which in the case of the transmitter is insulation breakdown. Next, there are a few other phases like channel selection, data preprocessing and labeling, feature engineering, and modeling, which will be described in detail in the subsequent sections.

Cooperation with the SMEs is crucial to learn the details about the components’ usage, health and failure patterns. The data scientists perform the research, and analysis based on the suggestions of SMEs and ensure that the data-driven approach is held. Moreover, the fact that the development of machine learning models requires specific knowledge and often quite a lot of experience leads to the fact that the models are developed by the data scientists with the support of SMEs, and not SMEs themselves.

The reason for developing the models in a data-driven approach, rather than a model-based approach, is that we can build models faster with less cost, using the historical data (Mosallam, 2014). Moreover, machine learning models provide efficiency and consistency, and are not dependent on the individual’s mistakes. Furthermore, the model-based approach requires a deep understanding of the physical mechanism of the failure, extensive experimentation, expert knowledge, and model verification, which is highly time-consuming (Mosallam, 2014).

4.1. Data Description

After each run, tool data consists of a hundred number of data channels generated at a record rate, which results in millions of data points from a single run. Only some of the channels hold valuable information about the health of a transmitter. Removing the channels that do not contain useful information about the health of a component should help to re-

duce the noise and, consequently, increase the algorithm efficiency. SME domain knowledge determines the choice of channels that hold relevant information about the health status. Depending on the collar size of the tool, there can be a single type of transmitter (monopole & quadrupole) or separate monopole and quadrupole transmitters used during the run. For each transmitter, data is stored in two firing voltage channels with positive polarity and negative polarity, from which data can be collected from both or a single channel, depending on the operation mode. Based on those channels the features should be created to distinguish between a healthy and a faulty transmitter, and used to build the fault detection model. Taking into consideration that there are different types of transmitters (monopole & quadrupole) and that they differ between multiple collar sizes of the tool, in the subsequent sections, we are going to cover the details for each size and type of transmitter.

4.2. Data Preprocessing

Once the tool is initialized and put in the well, the onboard system records measurements of transmitters’ voltage every 10 seconds. This data is available once the job is done and the tool is back to the surface. During the run, there can be situations when the transmitter is not firing, which affects in the measurement of 0 Volts. Such observations should be filtered out and not taken into consideration in the analysis. Figure 3 presents the time series of the transmitter’s raw voltage and voltage after preprocessing.

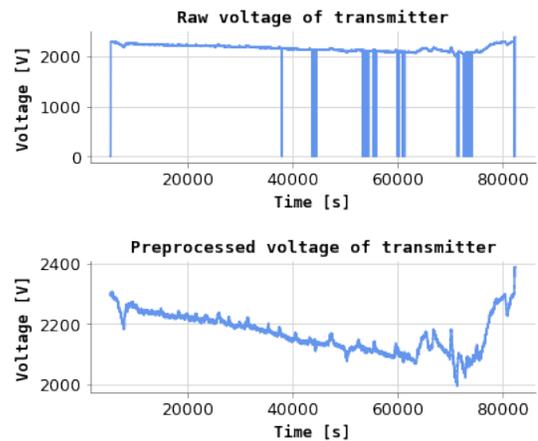


Figure 3. Transmitter’s raw voltage and preprocessed voltage

4.3. Data Labeling

The step of labeling is crucial for the successive steps of feature engineering and modeling. This phase is performed by the SMEs who, using their domain knowledge and experience, and taking into consideration the available status of the run, label the transmitter as healthy or faulty.

4.4. Feature engineering

The goal of this step is to transform the original channels raw data into features that represent the transmitter's health after each run in a statistical way. Performing this phase requires a good understanding of the failure mode. It's necessary to know the symptoms, as well as how the whole subsystem works and how transmitters cooperate within the system. Based on the collar size of a tool and transmitter type, the features can be extracted in different ways, depending on how the subsystem works. For each transmitter, to extract the features for modeling, we use data from one voltage channel. The decision if we use positive or negative polarity is made after consultation with SMEs about the operation mode for a certain tool type.

Tool A: Monopole & Quadrupole Transmitters For this collar size of the tool, four single monopole & quadrupole transmitters are used during the same run. The behavior and firing of transmitters are similar throughout the run if each of them is healthy, which can be seen in Figure 4. Based on the analysis of the dataset, during the run, one or more transmitters can fail (but there is no run in the available data where all of them failed together). Figure 5 shows the transmitters voltages for the run when Transmitter 3 has failed. There is a visible drop-down in the voltage of that transmitter, while the other transmitters' voltages remain similar. The analysis showed that the drop-down in the failed transmitter's voltage differs between different failures. Therefore the features will be extracted, using the co-dependency of the transmitters, not simply taking into consideration a single transmitter's voltage.

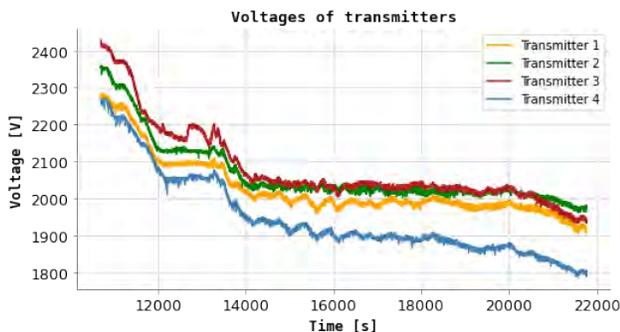


Figure 4. Tool A transmitters' voltages

Let T_i for $i \in \{1, 2, 3, 4\}$ denote a time series of voltage for transmitter used in a selected run. Let n denote the duration time (in seconds) of a run and $s \in \{1, \dots, n\}$ denote the moment in the run. Consequently,

$$T_i = [t_1, \dots, t_n] \quad (1)$$

is a vector of voltages for i -th transmitter within the selected

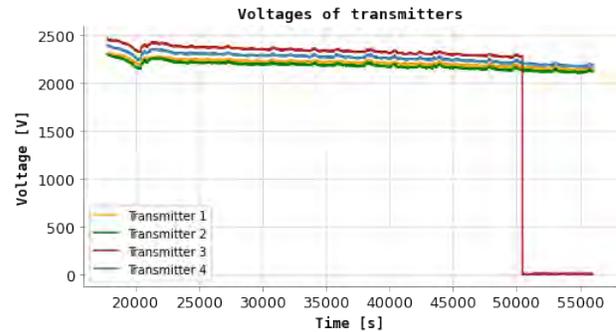


Figure 5. Tool A transmitters' voltages with one failed case

run. We calculate the features taking into consideration the maximum difference between the transmitters' voltages to differentiate the failed cases from healthy ones. For failed transmitters, depending on how many transmitters fail at once, at least one maximum difference will be high (as mentioned above, there were no runs for which we observed failures of all the transmitters). The following formula is used to calculate the feature x_k for $k \in \{1, 2, 3\}$, for the transmitter T_i :

$$x_k = \max_{s=1, \dots, n} |T_i - T_j|_s, \quad (2)$$

where $j \in \{1, 2, 3, 4\}$ and $j \neq i$. As a result, we get three dimensional space of features for each transmitter.

Tool B & C: Monopole & Quadrupole Transmitters In these tool types the monopole and quadrupole transmitters, are separate components. Four quadrupole and one monopole transmitters are used during the run. The reasons for creating features differently than for the transmitters of Tool A are the following:

- Like the transmitters described above, quadrupole behavior and firing are similar throughout the run if each is healthy, which can be seen in Figure 6. But we can observe that the firing mode is different from Tool A. It is noisier and the voltage can change drastically within 10 seconds (the spikes do not have to appear exactly at the same moment for all the transmitters).
- Monopole transmitters' voltage is similar to the voltage of transmitters of Tool A. Despite that, the fact that a single monopole transmitter is used within the tool requires different feature creation.

In the development phase, multiple statistical measures were checked such as a correlation between channels, standard deviation, minimum and maximum voltage, and so on, to highlight those that best separate healthy and faulty transmitters. Let T_i be defined by Equation (1). The following formula is

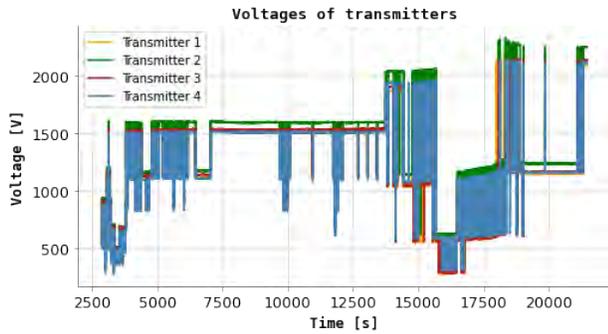


Figure 6. Tool B quadrupole transmitters' voltages

used to calculate the feature x for the transmitter T_i :

$$x = \min_{s=1, \dots, n} t_s, \quad (3)$$

4.5. Modeling

This phase associate the features created for each transmitter with the labels assigned by SMEs to map the relation between them (x, Y) where:

- for a single monopole & quadrupole transmitter from Tool A:

$$x = [x_1, x_2, x_3], \quad (4)$$

where x_i for $i \in \{1, 2, 3\}$ is defined by the formula (2).

- for both monopole and quadrupole transmitters from Tool B and Tool C: x is defined by the formula (3)

and for all kinds of transmitters Y is given by the formula:

$$Y = \begin{cases} 1, & \text{when transmitter is failed} \\ 0, & \text{when transmitter is healthy} \end{cases} \quad (5)$$

Features and corresponding labels are used to develop the models. Models are trained separately for the following types of transmitters:

- Tool A monopole & quadrupole transmitters
- Tool B quadrupole transmitters
- Tool C quadrupole transmitters
- Tool B & C monopole transmitters

To determine if the transmitter is healthy or faulty, we used the classification model. For each type of transmitter support vector machine (SVM) model with linear kernel was trained. In SVM, a hyperplane is constructed in n -dimensional space, where n is the number of features used in the model, in a way that best separates healthy and faulty classes (Hastie, Tibshirani, & Friedman, 2009).

Due to imbalanced class distribution, the evaluation of the model is performed using both the Accuracy and F1-score

(Fernández et al., 2018). The classification metrics are calculated as follows:

$$F1 = 2 * \frac{precision \cdot recall}{precision + recall} \quad (6)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

where

$$precision = 2 * \frac{TP}{TP + FP} \quad (8)$$

$$recall = 2 * \frac{TP}{TP + FN} \quad (9)$$

and

- TP stands for *True Positive* which is the number of the transmitters that model correctly classified as healthy,
- TN stands for *True Negative* which is the number of the transmitters that model correctly classified as faulty,
- FP stands for *False Positive* which is the number of the transmitters that are classified as healthy when they are faulty,
- FN stands for *False Negative* that is the number of the transmitters classified as faulty when they are healthy.

The number of transmitters (with regards to healthy and faulty) used to develop the fault detection models is presented in the Table 1.

Table 1. Number of healthy and faulty transmitters per transmitter type

Transmitter Type	Healthy	Faulty
Tool A monopole & quadrupole transmitters	68	8
Tool B quadrupole transmitters	43	9
Tool C quadrupole transmitters	71	17
Tool B & C monopole transmitters	31	10

Figures 7, 8, 9, 10 present trained SVM models with corresponding hyperplanes (in 1-dimensional and 3-dimensional spaces depending on the features used for modeling). Healthy transmitters are indicated by green points and faulty by red ones. It is visible that all types of transmitters classes are well separated, which is a good premise for prediction.

Before the SVM classification models were developed, simple thresholds were used to differentiate between healthy and faulty transmitters for all transmitters types. One of the significant aspects of model deployment is constant model improvement and evaluation. Therefore, the models should be automatically retrained on a scheduled basis so that any new phenomenon in the data could be incorporated and the SVM hyperplane could be moved. Due to that, it was decided to

deploy the SVM models instead of setting simple thresholds which would have to be manually reworked after some time.

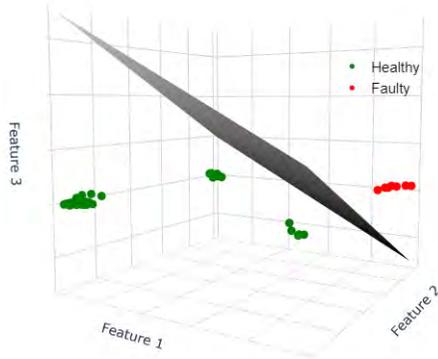


Figure 7. SVM hyperplane separating healthy and faulty monopole & quadrupole transmitters of Tool A

For Tool A monopole & quadrupole transmitters we observe three clusters of healthy transmitters. This phenomenon appears simply because we calculate the features in a certain order as it is mentioned in Equation (2). For example, for the Transmitter 1:

$$x_1 = \max_{s=1,\dots,n} |T_1 - T_2|_s,$$

$$x_2 = \max_{s=1,\dots,n} |T_1 - T_3|_s,$$

$$x_3 = \max_{s=1,\dots,n} |T_1 - T_4|_s,$$

while for the Transmitter 2:

$$x_1 = \max_{s=1,\dots,n} |T_2 - T_1|_s,$$

$$x_2 = \max_{s=1,\dots,n} |T_2 - T_3|_s,$$

$$x_3 = \max_{s=1,\dots,n} |T_2 - T_4|_s.$$

Therefore, the healthy cases are not grouped together. Looking at the three healthy clusters, we could conclude even more than the fact that a certain transmitter is healthy. We could use it to precisely say how many transmitters failed during the particular run (1, 2, or 3). However, for now, we use this model as a fault detection one, but there is a potential to be used in the future as a diagnostics model.

Model Performance Each machine learning model should be evaluated to check its performance and ability to generalize the learned pattern. Because we have highly limited amounts of data (see Table 1), a leave one out cross-validation method (LOOCV) was applied. This approach is a special case of K-fold cross-validation where the number of folds equals the number of transmitters that we have in the dataset.

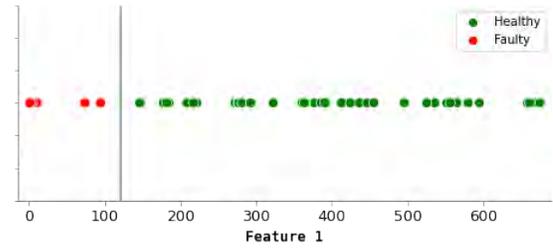


Figure 8. SVM hyperplane separating healthy and faulty quadrupole transmitters of Tool B

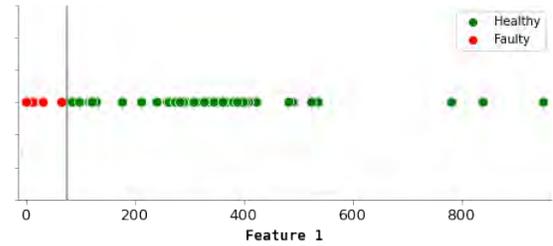


Figure 9. SVM hyperplane separating healthy and faulty quadrupole transmitters of Tool C

Thus, the algorithm is applied once for each transmitter, using the rest of the transmitters as a training dataset and adopting the selected one as a single test dataset. This method provides a low biased test accuracy and F1-score compared to using a single test dataset (Witten, Frank, & Hall, 2011). Table 2 presents the outcomes of the models’ performance based on LOOCV.

The models show high confidence with zero misclassification for Tool A monopole & quadrupole transmitters and Tool B quadrupole transmitters and only one misclassification for Tool C quadrupole transmitters and Tool B&C monopole transmitters, and hence high accuracy and F1-Score for each one of them. It is worth mentioning that the performances of the models are very high due to good feature engineering, not only thanks to the SVM model. Before the SVM models were chosen to be deployed, they were compared to other algorithms, to choose the best performing and most beneficial one. Performance and benefits of using the SVM model were compared for example to the usage of logistics regression. The results were the same for the 1-dimensional space, but SVM outperformed the logistics regression for the 3-dimensional space. Therefore, it was decided to choose the SVM models to be implemented.

5. CONCLUSION

This paper presents a data-driven fault detection method for transmitters. Characteristics of the different types of transmitters were detected and analyzed in the exploratory data analysis phase. Each transmitter has two voltage channels that are

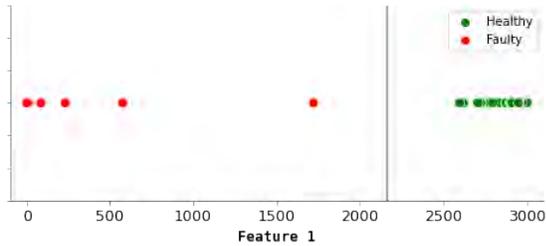


Figure 10. SVM hyperplane separating healthy and faulty monopole transmitters of Tool B&C

Table 2. Model performance metrics per transmitter type

Transmitter Type	Accuracy	F1-Score
Tool A monopole & quadrupole transmitters	100%	100%
Tool B quadrupole transmitters	100%	100%
Tool C quadrupole transmitters	99%	97%
Tool B & C monopole transmitters	98%	98%

used to construct representative features to identify the health status of a component. The classification model training is performed using these features to ensure the predictive power of the model, which is then used by the engineers and maintenance teams after each run to validate if the transmitter is healthy or not. The model performance validation resulted in a high F1-Score, which shows that the health of a transmitter can be identified correctly.

The proposed solution is deployed in the application that can be directly used by the field engineers and maintenance team to organize and plan their work more efficiently. The models and their performance are going to be constantly monitored and tested, and further improved if needed. Additional work is planned to build the diagnostics and prognostics models for the transmitters. However, several challenges such as the low historical data availability and uncertainty about the incipient failure mode of the components need to be considered and reworked.

REFERENCES

Fernández, A., García, S., Galar, M., Prati, R., Krawczyk, B., & Herrera, F. (2018). Performance measures. In *Learning from imbalanced data sets*. Springer, Cham.

Ganesh Kumar, D., Insozhan, N., & Parthasarathy, V. (2019). Recognition of faulty node detection using fuzzy logic in iot. *International Journal of Scientific and Technology Research*, 8(12), 1112 – 1116.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). Support vector machines and flexible discriminants. In *The elements of statistical learning*. Springer, New York, NY.

Kirschbaum, L., Roman, D., Singh, G., Bruns, J., Robu, V., & Flynn, D. (2020). AI-driven maintenance support for downhole tools and electronics operated in dynamic drilling environments. *IEEE Access*, 8, 78683-78701. doi: 10.1109/ACCESS.2020.2990152

Liu, C., Chen, B., Zhang, H., & Wang, X. (2018). Fault diagnosis application of short wave transmitter based on stacked auto-encoder. In *2018 IEEE 4th international conference on computer and communications (iccc)* (p. 119-123).

Liu, S., Xu, L., Li, Q., Zhao, X., & Li, D. (2018). Fault diagnosis of water quality monitoring devices based on multiclass support vector machines and rule-based decision trees. *IEEE Access*, 6, 22184-22195. doi: 10.1109/ACCESS.2018.2800530

Mosallam, A. (2014). *Remaining useful life estimation of critical components based on bayesian approaches*. (PhD dissertation). Université de Franche-Comté.

Mosallam, A., Laval, L., Youssef, F. B., Fulton, J., & Viasolo, D. (2018). Data-driven fault detection for neutron generator subsystem in multifunction logging-while-drilling service. In *PHM society european conference*.

Mosallam, A., Medjaher, K., & Zerhouni, N. (2016). Data-driven prognostic method based on bayesian approaches for direct remaining useful life prediction. *Journal of Intelligent Manufacturing*, 27, 1037–1048. doi: 10.1007/s10845-014-0933-4

O'Connor, P., & Kleyner, A. (2012). *Practical reliability engineering, fifth edition*. John Wiley & Sons, Ltd.

Tugova, E. S., Salov, D. D., & Bushuev, O. Y. (2021). Diagnostics of a pressure transmitter based on output signal noise characteristics. In A. A. Radionov & V. R. Gasiyarov (Eds.), *Proceedings of the 6th international conference on industrial engineering (icie 2020)* (pp. 1298–1307). Springer International Publishing.

Vachtsevanos, G., Lewis, F. L., Roemer, M., Hess, A., & Wu, B. (2006). Intelligent fault diagnosis and prognosis for engineering systems. In *1st ed. hoboken*.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

BIOGRAPHIES



Karolina Sobczak-Oramus is a data scientist at Schlumberger. She holds a Master of Science in Mathematics from the Jagiellonian University of Kraków, Poland. Her main research interests are in the fields of machine learning, artificial intelligence and data mining.



Ahmed Mosallam is an analytics manager at Schlumberger technology center in Clamart, France. He has his PhD degree in automatic control in the field of PHM from the University of Franche-Comté. His main research interests are signal processing, data mining, machine learning and PHM.



Caner Basci is a senior electrical sustaining engineer at Schlumberger. He holds an M.S. in the field of Very Large Integrated Circuits about Complementary Metal Oxide Semiconductor Image Sensors in Electrical Engineering Department at the University of

Tokyo, Japan. His main interests are Power Electronics Circuits, Analog Circuit Design, Data Processing and Oil & Gas Exploration with Acoustics.



Jinlong Kang is a PhD student at University of Franche-Comté in Besançon and a data scientist at Schlumberger technology center in Clamart, France. He holds a B.S. in Industrial Engineering and an M.S. in Mechanical Engineering both from University of Electronic Science and Technology of China. His main research interests are Prognostic and Health Management, maintenance decision-making, data mining and machine learning.

Autonomous Bearing Tone Tracking Algorithm

Alon Sol¹, Eyal Madar¹, Jacob Bortman¹, Renata Klein²

¹ PHM Laboratory, Department of Mechanical Engineering, Ben-Gurion University of the Negev, P.O.B

653, Beer-Sheva 8410501, Israel

sola@post.bgu.ac.il

evalmad@post.bgu.ac.il

jacbori@post.bgu.ac.il

²R.K. Diagnostics, P.O. Box 101, Gilon, D.N. Misgav 20103, Israel

Renata.Klein@rkdiagnostics.co.il

ABSTRACT

To date, much of the research done in the field of condition monitoring of rotating machinery is conducted in the frequency domain. The frequency domain analysis specifically for bearings is based on extracting features from the spectrum of the vibration signature. These features are mostly based on the amplitude at the bearing tones along with their sidebands and high order harmonics. Therefore, it is important to determine the location of the mentioned bearing tones in the spectrum accurately and automatically. For the case of ball bearings this process can be problematic due to slippage of the rolling elements and variations in the angle of contact. These may cause the bearing tone to deviate from its nominal value.

To this day, the common practice for bearing diagnostics is based on the vibration level at the analytical bearing tones or involvement of experts to identify the true location of the bearing tone. In this research an autonomous algorithm for bearing tone extraction, based on pattern matching, was developed. The proposed algorithm is based on the common assumption that the spectrum of a faulted bearing contains a certain known pattern of prominent peaks. The algorithm “scans” the entire spectrum and determines the frequency value which has the highest correlation to the mentioned pattern.

The proposed algorithm was validated and its capabilities are illustrated using experimental data. This algorithm is able to assist any diagnostic approach towards automatic and reliable feature extraction process, both for physics based and data driven approaches.

Alon Sol et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

Condition based monitoring (CBM) has become a dominant approach in maintenance as it can save resources and manpower while improving safety. The general concept of CBM is monitoring the machine's components and estimating its current health condition.

This research focuses on the field of rotating machines, and specifically in the field of bearings since bearings are highly prone to faults. A fault in a bearing can cause malfunction of the entire machine, leading to failure and environmental damage (Randall, 2011).

One of the common approaches for monitoring the condition of bearings is vibration analysis. This approach is based on the assumption that a faulted bearing will generate an impulse followed by a dynamic response each time a component encounter with the fault. By analyzing this vibration signature, the fault severity can be estimated (Malla and Panigrahi, 2019).

Vibration analysis in the frequency domain is one of the common approaches in the field of bearings. This approach assumes that the spectrum of a healthy bearing will be relatively “clean” while a faulted bearing has unique prominent peaks in the spectrum according to the rate of the impulses. These frequencies are named bearing tones which can be calculated based on kinematic equations. This assumption was firstly modelled and discussed by (McFadden and Smith, 1984).

Most frequency domain analysis methods are based on manipulating the spectrum around the bearing tones. Zhang et al. reviewed several diagnostic approaches and demonstrated a method which tracks the amplitude of the peak at the bearing tone and utilizes it to estimate the fault size, which is related to the bearing condition (Zhang et al., 2022). There is a great need to determine the precise location

of the bearing tone automatically. This demand applies for physics-based and/or data-based approaches which utilize manipulations around the peaks for feature extraction. For instance, Nissim et al. used features which are based on the location of the bearing tones (Nissim et al., 2021). These approaches are highly sensitive to the location of the analyzed peaks as it may affect the results. Therefore, the accuracy of the peak location is important.

For the best of our knowledge, only limited research provides algorithms for this purpose. A paper which was published by (Kass, Raad, and Antoni, 2019) displayed an algorithm for bearing tone detection. It is aimed to detect small deviations of the bearing tone due to slippage of the rolling elements. In many applications these deviations are significantly larger than those caused by slippage. This study proposes a new algorithm for bearing tone determination which can handle large deviations in noisy spectrum. The algorithm can assist frequency domain diagnostic approaches.

The article is organized as follows. Chapter 2 presents the required background for the understanding and establishment of the suggested algorithm. Chapter 3 presents the algorithm inputs and procedure followed by a demonstration in Chapter 4. Lastly, Chapter 5 summarizes the work.

2. BACKGROUND

A bearing contains four components, an inner race, an outer race, rolling elements and a cage. In the case of a fault in one of these components a specific frequency, named bearing tone, will rise in the spectrum. The bearing tone can be calculated analytically based on kinematic equations. BPFO is the ball pass frequency on the outer race, BPFI is the ball pass frequency on the inner race, FTF is the fundamental train frequency and BSF is the ball spin frequency.

$$BPFO = \frac{f_r}{2} \cdot n \cdot \left[1 - \left(\frac{B_d}{P_d} \right) \cdot \cos(\beta) \right] \quad (1)$$

$$BPFI = \frac{f_r}{2} \cdot n \cdot \left[1 + \left(\frac{B_d}{P_d} \right) \cdot \cos(\beta) \right] \quad (2)$$

$$FTF = \frac{f_r}{2} \cdot \left[1 - \left(\frac{B_d}{P_d} \right) \cdot \cos(\beta) \right] \quad (3)$$

$$BSF = f_r \cdot \frac{P_d}{B_d} \cdot \left[1 - \left(\frac{B_d}{P_d} \right)^2 \cdot \cos(\beta)^2 \right] \quad (4)$$

Where f_r is the rotating speed, n is the number of balls, B_d is the ball diameter, P_d is the pitch diameter and β is the angle of contact.

A healthy bearing should not generate any impulses in the vibration signature and consequently no prominent peaks are expected in the spectrum while the spectrum of a faulted bearing will contain prominent peaks at the bearing tones. Due to the Fourier transform properties, these peaks will appear along with their harmonics.

Another phenomenon is the Amplitude Modulation (AM), which is caused by load variations (loading zone or unbalance of the shaft). This phenomenon occurs when a signal with frequency f_{tone} is modulated by a signal a frequency f_m , usually the bearing tone will be modulated by the shaft rotation frequency. Therefore, in addition to the bearing tone peaks we can expect sidebands in f_m distance from the bearing tone as a result of AM. These sidebands will also appear with their harmonics, e.g., $f_{tone} \pm f_m, f_{tone} \pm 2f_m, \dots, f_{tone} \pm n f_m$.

Overall, assuming a faulted bearing with a bearing tone of f_{tone} and a modulating frequency of f_m , the pattern illustrated in Figure 1 is expected to appear in the spectrum. This pattern only includes two harmonics with three sidebands and may include many more.

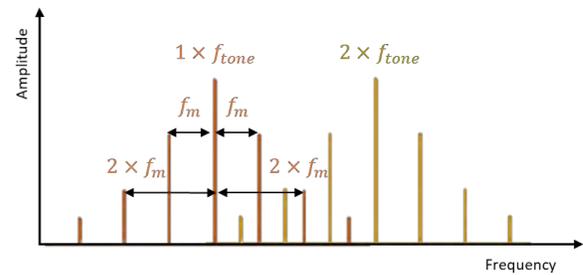


Figure 1. Expected pattern of a faulted bearing.

As mentioned before, many approaches utilize the spectrum around its bearing tones in order to acquire information about the bearing health condition. Eq. (1)-(4) are based on two assumptions which do not necessarily apply for real cases and might cause the real bearing tone to deviate from its analytical value.

The first assumption is rolling without slippage. The slippage of the rolling elements in the bearing can be explained by the fact that each rolling element is loaded differently depending on its position. As a result, each element tries to rotate at a different velocity yet the cage forces them to remain in equal distance by causing slippage (Randall, Antoni, and Chobsaard, 2001). Another research conducted by (Pennacchi et al., 2011) concludes that the amount of slippage in bearing may change according to the operating conditions, bearing type and the bearing tone of interest. For instance, the BPFI will be more affected by slippage than the BPFO. Another assumption is that the angle of contact β is constant at the nominal value. In many cases the angle of contact varies from its nominal value depending on the ratio between the axial and radial loads and operating conditions. These changes of the angle of contact might change the bearing tone significantly as can be deduced from Eq. (1)-(4). Therefore, it is required to locate the true bearing tones in every vibration record. When the change of the bearing tone is large, e.g., for large variations in the angle of contact, or when there are

many peaks in the vicinity of the bearing tone, the task becomes more challenging.

3. BEARING TONES LOCATION ALGORITHM

The proposed algorithm Bearing Tones Location (BTL) is based on pattern matching, since it is assumed that the spectrum of a faulted bearing will contain prominent peaks which match a pattern similar to the illustration in Figure 1.

Since the spectrum may include many peaks related to other components (shafts, gears, etc.), the BTL search for the defined pattern simultaneously which aids to avoid irrelevant peaks. In cases where the spectrum contains numerous peaks related to other components, filtering methods can be applied to remove the discrete frequencies from shafts, gears and rotors.

The BTL “scans” the entire spectrum and finds the frequency value which has the highest correlation to the expected pattern. Once the bearing tone is determined, the BTL calculates the result score by evaluating the statistical distance of the whole pattern based on the selected bearing tone to the alternative patterns. The higher the distance the more reliable the determination is.

Since the location of the fault is unknown, the BTL is designed to determine different bearing tones using their corresponding patterns, each pattern includes a different set of harmonics and sidebands corresponding to different failure modes, e.g., defect on outer race or inner race or ball, etc. It is important to note that the BTL will produce a meaningful result only if such pattern exists in the spectrum, otherwise the score (statistical distance from the alternatives) will be low.

The process of scanning the spectrum for a pattern can be conducted in two ways. The first way is searching for the whole pattern, from the first harmonic to a high harmonic defined as an input to the algorithm. The second way is by scanning different sets of harmonics each time. For each set, the BTL determines the most probable location of the bearing tone for this specific set, and the final bearing tone is determined to be the location found in the set which produced the highest total amplitude. The second scanning process helps to detect the bearing tone in case a small number of peaks are prominent in the spectrum.

There is a tradeoff which needs to be taken in account when defining the pattern for the BTL. When the pattern contains more harmonics and more sidebands, the pattern gets richer and the result will be more reliable since the odds for accidental high correlation between the pattern and the spectrum decreases. If the pattern is too rich, meaning that it includes more harmonics and sidebands than what rises in the spectrum, the score might be affected since it will sum a lot of irrelevant bins. The recommended pattern should include the minimal amount of harmonics and sidebands which can provide a reliable result.

For visualization, in Figure 2 two arbitrary frequency values were examined and matched to a pattern consisting of one harmonic with one sideband. It can be seen that by summing the amplitude of the spectrum matching the red pattern the total corresponding amplitude is higher than the case of the green pattern, in this example the red pattern is more likely to represent the bearing tone.

The spectrum of the vibration signature is calculated using DFT, hence the frequency axis is discrete and divided into “bins”. Since the frequency resolution is constant, each bin in the first harmonic corresponds to n bins in the n^{th} harmonic. Therefore, to determine the bearing tone with the highest accuracy, for every approach, it is best to find the highest harmonic and divide it by the corresponding harmonic number.

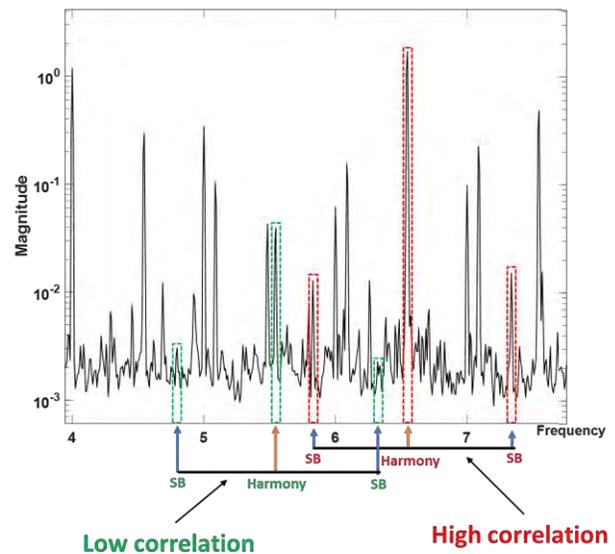


Figure 2. General pattern matching example.

3.1. Algorithm Inputs

In the following subchapter, the BTL inputs will be presented and explained. Note that for the next section, the term “harmonic” refers to the harmonic and its corresponding sidebands.

- Analytical bearing tone

The kinematic bearing tone can be calculated using the rotating speed and the geometry of the bearing using Eq. (1)-(4).

- Range of the first harmonic

Since the experimental bearing tone may deviate from the analytical value, this input defines the search range around the analytical bearing tone. To set this input properly a preliminary evaluation of the variations of the angle of

contact β and changes due to slippage are required. The search range is not necessarily symmetrical and has no limitation, but it may affect the score and run-time of the algorithm.

For bearings with no axial load the only cause for deviation is slippage, therefore a search range of 1% is mostly sufficient. For the case of a bearing with axial load the search range should increase according to the possible variations of the contact angle.

- Number of harmonics in the set

This input determines the number of harmonics in the set that the algorithm will scan. The number of harmonics has to be bigger than one.

- Maximum harmonic

If the chosen scanning approach is by using different sets of harmonics, then this input determines the highest harmonic in the search range. This number is equal or larger than the number of harmonics in the set. This value affects the different patterns that the algorithm scans, For example if the maximum harmonic is 10 and the number of harmonics in the pattern is 4, the algorithm will scan the following - harmonics 1:4,2:5,...,7:10.

- Modulating frequency

The modulating frequency determines the distance of the sidebands expected to rise around the bearing tone harmonics. In most cases the amplitude modulation will occur due to shaft unbalance or fluctuations of the load, meaning that the modulating frequency is usually the shaft rotation frequency.

- Number of sidebands

This input determines the number of sidebands around each harmonic in the expected pattern that the algorithm will scan. The higher the number the more specific the pattern will be. For most cases 1-3 sidebands is sufficient to create a specific enough pattern.

3.2. Algorithm Procedure

The general scheme of the algorithm is presented in Figure 3 as a block diagram, each step of the BTL will be explained throughout this subchapter. The procedure will be described according to the general case of using different sets. When the number of harmonics in the set is equal to the maximum number of harmonics the algorithm works on one set.

First, the BTL builds the desired pattern to search for and defines the search range and the sets of harmonics.

Since high resolution is desired, the BTL will find the frequency bins which correspond to the deviation range around the highest harmonic in the specific set, these will be noted as $bins = \{i \in 1,2,3 \dots N\}$.

For every bin i in this range the BTL will find the matching bins of lower harmonics and their associated sidebands and will sum their amplitudes. This step is conducted for each harmonic in the defined set, summing all of their amplitudes to the matching bin which will result in an amplitude A_i .

Once this procedure is done for each bin, the bearing tone is determined to be the bin which corresponds to the highest total amplitude divided by the highest harmonic number n in the set as can be seen in Eq. (5).

$$bearing\ tone = \frac{\operatorname{argmax}(A_i)}{n} \quad (5)$$

After the bearing tone is determined, the score of the result is estimated by calculating the Z Score of the amplitude of the chosen bin according to Eq. (6).

$$ZS = \frac{\max_i(A_i) - \mu}{\sigma} \quad (6)$$

Where μ is the average and σ is the standard deviation of the sum of amplitudes in the different bins, respectively. The Z-Score measures the prominence of the amplitude corresponding to the chosen bin relative to all other bins. This correlates to the distance of the chosen bin amplitude from the mean amplitudes σ , thus correlates to the confidence of the result.

After the bearing tone and the result score value are found according to the current set of harmonics, the same procedure is conducted using the next set which includes different harmonics. After the procedure is done for each set, the bearing tone which has the highest amplitude value is chosen.

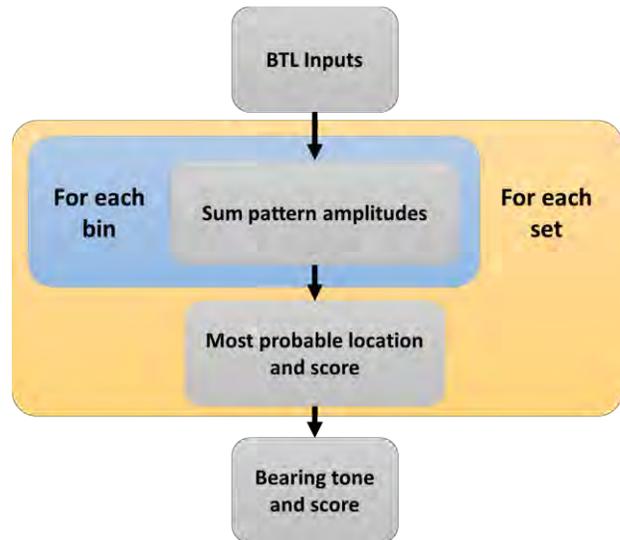


Figure 3. Algorithm scheme

4. DEMONSTRATION

To demonstrate the algorithm efficiency and to further explain the required inputs, two experiments were conducted. Throughout each experiment the vibration signature was recorded. Each vibration recording was processed according to the methodology presented in Figure 4. Eventually, the spectrum of the recording was obtained in the order domain instead of the frequency domain by calculating the spectrum after angular resampling. The angular resampling corrects the effect of fluctuations in rotating speed since the bearing tone will be normalized to cycles instead of time. Then the envelope of the signal is calculated to emphasize the bearing related phenomena. The final step is estimating the Power Spectral Density (PSD) of the envelope. These processing steps are not mandatory and the BTL can be applied on any spectrum in the order/frequency domain. Yet, when applying it in the frequency domain the peaks will be smeared on several bins which might affect the accuracy of the result.

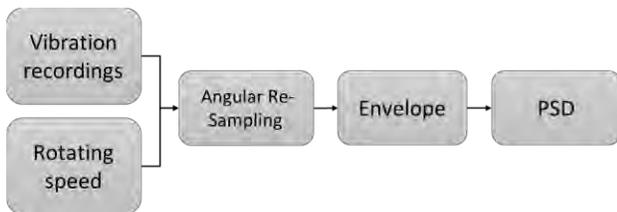


Figure 4. Signal processing methodology.

For the first example the test rig included a deep groove bearing fitted on a shaft, rotated in 40 [Hz]. The bearing was being radially loaded by 200 [N] flywheels and was artificially induced by a fault on the outer race. After the signal processing procedure was applied, the spectrum obtained is presented in Figure 5 in the order domain.

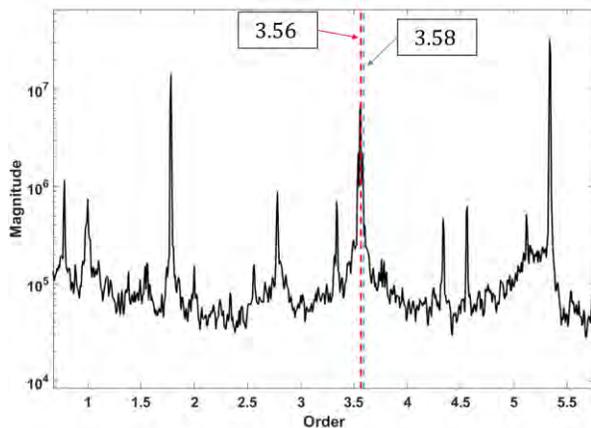


Figure 5. First example of the algorithm.

Using the geometric parameters of the bearing the analytical BPFO bearing tone is 3.586 *order* and is presented in a blue dashed line in . The algorithm was applied with inputs of the analytical bearing tone, a 1% of the analytical bearing tone as

search range, 8 harmonics and 1 sideband of 0.4 order (FTF). The bearing tone was determined to be 3.5625 *order* as can be seen in a red dashed line with a deviation of 0.64% from the analytical bearing tone, with a Z Score of 8.52.

For the second example an endurance test was conducted, the test rig included an angular contact bearing fitted on a shaft rotating in a constant speed of 166 [Hz] and loaded axially by 5500 [N]. A random recording was processed using the procedure presented and the obtained spectrum is presented in Figure 6. Using the geometric parameters of the bearing the analytical BPFI bearing tone is 6.58 *order* and is represented by a blue dashed line in Figure 6. The algorithm was applied with inputs of the analytical bearing tone, a 1.5% of the analytical bearing tone as deviation range, 10 harmonics and 2 sidebands of 1 order (shaft rotation). The bearing tone was determined to be 6.51 *order* as can be seen in a red dashed line in Figure 6, its sidebands are marked with red dotted lines. Observing the spectrum, it can be seen that even though it is relatively clean around the first harmonic and the determination of the bearing tone is simple, the deviation from the analytical bearing tone is significant at 1.06%. The Z Score of this result is 9.72 making the determination considerably confident.

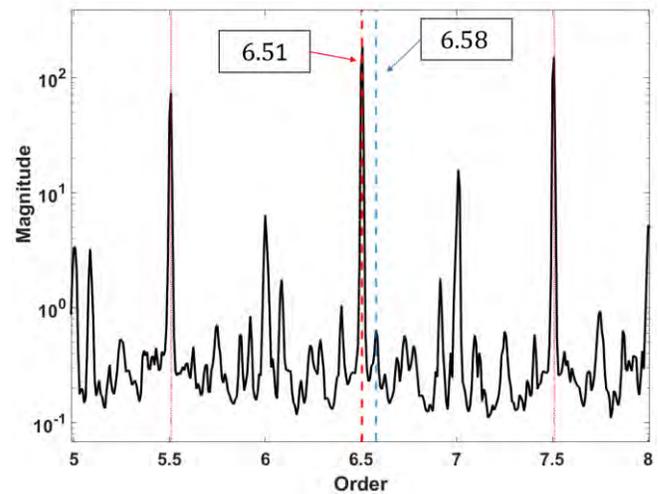


Figure 6. Example of the algorithm result.

It is also noticed that by using the analytical bearing tone for further analysis the results will not reflect the amplitudes of the peaks excited by the faulty bearing. Even though the task of locating the bearing tone in the first harmonic is quite simple, the resolution gained from this process is limited and can also affect any further analysis. The BTL was applied on many more recordings and was thoroughly validated with various fault cases and operating conditions and is successfully capable of determining the bearing tone.

To further present the complexity of locating the bearing tone at higher resolution, the same spectrum is presented around the 10th harmonic in Figure 7. As can be seen, the spectrum

presents many peaks which represent different harmonics and their corresponding sidebands. The 9th harmonic is marked with a blue dashed line and its sidebands are marked with blue dotted lines, same goes for the 11th harmonic marked in green. The 10th harmonic which we desire is marked in red. In high orders, sidebands of different harmonics can overlap one another and cause confusion making the task of determining the bearing tone becomes very complicated, emphasizing the need to search for the entire pattern match. The location of the 10th harmonic of analytical BPF bearing tone is marked in a light blue dashed line and its sidebands are marked as blue arrows above the spectrum.

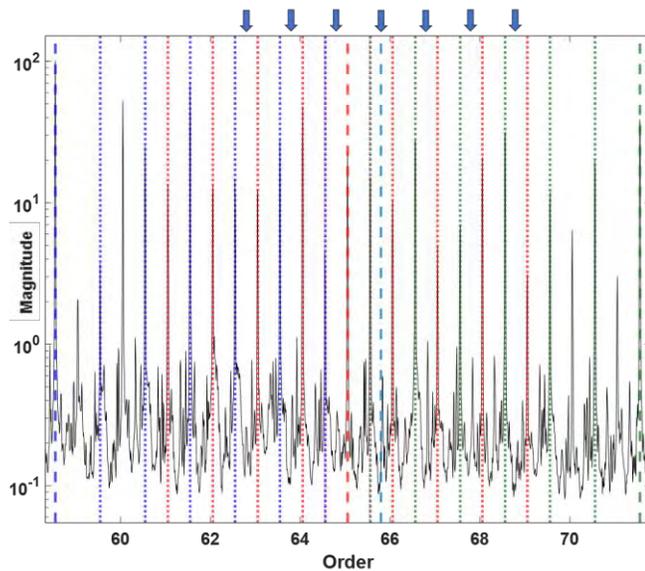


Figure 7. Second example with higher harmonics.

It is important to mention that the deviations from the analytical bearing tone will linearly increase when examining higher harmonics, which will affect any further analysis and may cause erroneous conclusions regarding the bearing health condition.

5. SUMMARY

In this article the BTL algorithm was introduced. It determines the bearing tone of a bearing with known geometry. It is based on the assumption that a faulted bearing will produce a pattern of prominent peaks in the spectrum of the vibration signal. This pattern is defined based on the bearing faulted component. The bearing tones, i.e., the frequencies of interest, might deviate from their nominal value since bearings are prone to slippage and load variations. These deviations can cause unreliable results of any further analysis. The BTL algorithm is able to determine the bearing tone automatically and accurately and evaluate the confidence of the determination.

Both the reliability and the uniqueness of the BTL algorithm result from scanning the entire spectrum for the specific pattern, which is assumed to rise in case of a fault. The BTL algorithm can be applied on every spectrum in the frequency or order domain with any pre-processing stages. The proposed algorithm was validated and its capabilities are illustrated using experimental challenging data. This algorithm is able to assist any diagnostic approach towards automatic and reliable feature extraction, both for physics based and data driven based approaches.

6. BIBLIOGRAPHY

- Kass, S., A. Raad, and J. Antoni. 2019. "Self-Running Fault Diagnosis Method for Rolling Element Bearing." *Mechanisms and Machine Science* 58(March):127–40. doi: 10.1007/978-3-319-89911-4_10.
- Malla, Chandrabhanu, and Isham Panigrahi. 2019. "Review of Condition Monitoring of Rolling Element Bearing Using Vibration Analysis and Other Techniques." *Journal of Vibration Engineering and Technologies* 7(4):407–14. doi: 10.1007/s42417-019-00119-y.
- McFadden, P. D., and J. D. Smith. 1984. "Model for the Vibration Produced by a Single Point Defect in a Rolling Element Bearing." *Journal of Sound and Vibration* 96(1):69–82. doi: 10.1016/0022-460X(84)90595-9.
- Nissim, Yonatan, Renata Klein, Jacob Bortman, and Jonathan Rosenblatt. 2021. "A Hybrid Method for Degradation Assessment and Fault Detection of Rolling Element Bearings." P. 84 in *enbis*. Online conference.
- Pennacchi, P., P. Borghesani, S. Chatterton, and R. Ricci. 2011. "An Experimental Based Assessment of the Deviation of the Bearing Characteristic Frequencies." *Proceedings of the 6th International Conference Acoustical and Vibratory Surveillance Methods and Diagnostic Techniques* 1–8.
- Randall, R. B., J. Antoni, and S. Chobsaard. 2001. "The Relationship between Spectral Correlation and Envelope Analysis in the Diagnostics of Bearing Faults and Other Cyclostationary Machine Signals." *Mechanical Systems and Signal Processing* 15(5):945–62. doi: 10.1006/mssp.2001.1415.
- Randall, RB. 2011. *Vibration-Based Condition Monitoring*. Vol. 11.
- Zhang, Hengcheng, Pietro Borghesani, Robert B. Randall, and Zhongxiao Peng. 2022. "A Benchmark of Measurement Approaches to Track the Natural Evolution of Spall Severity in Rolling Element Bearings." *Mechanical Systems and Signal Processing* 166(March 2021):108466. doi: 10.1016/j.ymsp.2021.108466.

Alon Sol received his B.Sc. degree in Mechanical engineering from Ben-Gurion University of the Negev. Currently, he is a M.Sc. student. His study focuses on vibration analysis of rolling element bearing. His main areas of research are signal processing, dynamic behavior of bearings and data analysis.

Eyal Madar received his B.Sc. and M.Sc. degree in Mechanical Engineering from Ben-Gurion University of the Negev. Currently, he is a Ph.D. student. His study focuses on characterization the dynamic behavior of faulted bearings by using physical based models vibration analysis. His main areas of research interest are dynamic modelling, signal processing and data analysis.

Dr. Renata Klein received her B.Sc. in Physics and Ph.D. in the field of Signal Processing from the Technion, Israel Institute of Technology. In the first 17 years of her professional career, she worked in ADA-Rafael, the Israeli Armament Development Authority, where she managed the Vibration Analysis department. In the decade that followed, she focused on development of vibration based health management systems for machinery. She invented and managed the development of vibration based diagnostics and prognostics systems that are used successfully in combat helicopters of the Israeli Air Force, in UAVs and in jet engines. Renata is a lecturer in the faculty of Aerospace Engineering of the Technion, and in the faculty of Mechanical Engineering in Ben Gurion University of the Negev. In the recent years, Renata is the CEO and owner of R.K. Diagnostics, providing R&D services and algorithms to companies who wish to integrate Machinery health management and prognostics capabilities in their products.

Prof. Jacob Bortman joined the academic faculty of Ben-Gurion University of the Negev in September 2010 as a full Professor. Prof. Bortman spent thirty years in the Israel Air Force (IAF), retiring with rank of Brigadier General. His areas of research in the Dept. of Mechanical Engineering include: Health usage monitoring systems (HUMS); Conditioned based maintenance (CBM); Usage and fatigue damage survey; Finite Element Method; and Composite materials.

Noise-robust representation for fault identification with limited data via data augmentation

Zahra Taghiyarrenani¹, Amirhossein Berenji²

¹ *Center for Applied Intelligence Systems Research, Halmstad University, Halmstad, Halland, 30118, Sweden*
zahra.taghiyarrenani@hh.se

² *Department of Mechanical and Energy Engineering, Shahid Beheshti University, Tehran, Tehran, 1983969411, Iran*
a.berenji@mail.sbu.ac.ir

ABSTRACT

Noise will be unavoidably present in the data collected from physical environments, regardless of how sophisticated the measurement equipment is. Furthermore, collecting enough faulty data is a challenge since operating industrial machines in faulty modes not only has severe consequences to the machine health, but also may affect collateral machinery critically, from health state point of view. In this paper, we propose a method of denoising with limited data for the purpose of fault identification. In addition, our method is capable of removing multiple levels of noise simultaneously. For this purpose, inspired by unsupervised contrastive learning, we first augment the data with multiple levels of noise. Later, we construct a new feature representation using Contrastive Loss. The last step is building a classifier on top of the learned representation; this classifier can detect various faults in noisy environments. The experiments on the SOUTHEAST UNIVERSITY (SEU) dataset of bearings confirm that our method can simultaneously remove multiple noise levels.

1. INTRODUCTION

Measurement noise is an integral part of instrumentation processes. It introduces noticeable amount of uncertainties, which complicates the decision making procedure. From the classification problem point of view, addition of noise results in severe reduction of separability between different classes, as it would scatter observations of different classes, which were fairly separable before addition of noise, all over the feature space. As it would result in poor classification performance, the employment of denoising techniques is an essential step in environments with significant level of noise presence.

Recently, Deep Learning has gained significant attention towards itself; however, coping with noise presence is still a challenge for Deep Learning-based methods (Liu, Zhou, Zhao, Shen, & Xiong, 2019). Despite of being highly admired due to their performance, deep learning methods are notorious for the requirement of huge amounts of information for training. Even with undeniable technological advancements during recent decades, it is still quite challenging to provide Deep Learning methods with sufficient training data; therefore, it is crucially important to employ strategies and techniques that make Deep Learning methods applicable in limited data scenarios. This matter comes to higher level of importance in the fields related to machinery health diagnosis, as running industrial pieces of equipment in faulty modes would bring up severe consequences (Wang et al., 2020).

Moreover, in a fault identification problem, different faults can be distinguished to some extent, but interference resulting from various factors, including measurement noise, inevitably weakens their separation. Therefore, achievement of acceptable separability of faults according to the set of features extracted, becomes a matter of great importance in the implementation of a fault identification model. Contrastive learning is a well established strategy to extract a feature space where different faults are properly discriminated in the constructed space (Le-Khac, Healy, & Smeaton, 2020). Being focused on the construction of a feature space where the distance between observations from similar classes (faults) is minimized, while clusters of different classes (faults) orient farthest from each other, approaches based on contrastive learning are discriminative feature extractors.

In this paper, we propose a new method for learning a new representation using contrastive learning and Siamese neural network for the denoising task. In addition to having noise-robustness and class discriminative properties, the proposed method addresses situations where enough labeled samples

Zahra Taghiyarrenani et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

are not available.

The rest of the paper is organized as follow: In section 2 the theoretical background about contrastive learning and Siamese neural networks are discussed. In section 3, we review the most recent and related works in denoising for fault identification. Section 4 defines the problem and the proposed method is presented in section 5. In Section 6, we describe experiments conducted to validate the effectiveness of the proposed method. Lastly, we will conclude our work in section 7.

2. BACKGROUND

2.1. Contrastive representation learning

In contrastive representation learning, an embedding space is constructed to close similar samples and separate dissimilar ones. In addition, the samples are paired, if two paired samples are similar we call the pair as a positive pair, otherwise a negative pair. To perform contrastive learning, it is, therefore, necessary to find similarities between samples. Moreover, contrastive learning can be accomplished either supervised or unsupervised. The similarity between samples is specified according to the labels of the samples in supervised contrastive learning. Therefore, in the constructed embedding by supervised contrastive learning, samples with the same label are placed near each other and those with different labels are placed far apart from each other. This approach is an example of few-shot learning methods (Jadon, 2020). Unsupervised contrastive learning incorporates data augmentation. In fact, it is assumed that the augmentation process will not change the class of a sample. Therefore, each sample and its augmented counterpart are considered to be similar and construct a positive pair, and consequently will be placed near to each other in the new constructed space. This approach is an example of a self-supervised learning methods (Le-Khac et al., 2020; Chen, Kornblith, Norouzi, & Hinton, 2020).

Different loss functions have been proposed in the literature to perform contrastive learning, including the Contrastive Loss, Triplet Loss, Lifted Structured Loss, etc (Le-Khac et al., 2020; Oh Song, Xiang, Jegelka, & Savarese, 2016). In this paper, we use Contrastive Loss function for training process.

Let X and \mathbf{X} represent input and embed spaces, respectively; $f : X \rightarrow \mathbf{X}$ is a function that maps the original space to the embed space. In addition, c is a contrastive label associated with a pair of samples. It is equal to zero if the pair of samples is similar, and to one if they are dissimilar. Equation 1 shows the Contrastive Loss.

$$ContrastiveLoss = (1 - c)D_w^2 + (c)(\max(0, m - D_w))^2 \quad (1)$$

where D_w is a similarity index, such as Euclidean Distance

and m is a parameter known as margin. Margin is supposed to be the distance between different classes, in the constructed feature space. According to the contrastive loss equation, the first term is supposed to represent similar observations as closely as possible, while the second term is supposed to increase the distance between dissimilar observations. (Jadon, 2020).

2.2. Siamese Neural Network

A siamese neural network consists of a dual and symmetric architecture, in which a pair of identical models are used to extract embedding corresponding to the given pairs of observations (Chicco, 2021). Given observations are either positive pairs (belonging to the same classes) or negative pairs (belonging to the different classes). During the training process, by comparing the observations available in training pairs, the network mines the input data. An architecture of a siamese neural network is shown in the figure 1c.

3. RELATED WORKS

Being capable of reconstructing a noise-free version of given noisy corrupted observations, Denoising Autoencoders (DAE) (Vincent, Larochelle, Bengio, & Manzagol, 2008) are highly used for fault diagnosis in noisy environments. For example, in (Zhao, Lu, Ma, & Wang, 2015), a hybrid classification approach consisting of an unsupervised feature learning using a Stacked Denoising Autoencoder and a consecutive fine-tuning using softmax regression is used for fault detection in bearing. Noise presence in this study is modeled by setting a randomly chosen fraction of the units in the input of the network to zero. Moreover, dropout is used during the training process of Denoising Autoencoders to not only prevent the network from overfitting, but also improve the robustness of the network towards noise presence. Similarly, in (Liu et al., 2019) the performances of a 1-D Convolutional Denoising Autoencoder to reconstruct noise free versions of given noisy observations and a conventional neural network to use the reconstructed version to identify the health state, is evaluated. This study uses a dual approach to take into account for noise presence. The presence of noise for cases involving training the denoising autoencoder is done by adding Gaussian noise with various signal to noise ratio, from -2 to 12 dB, while noise presence during the training of the conventional neural network is achieved by randomly setting units of input to zero and the rate of chosen units varies from 0.2 to 0.8. The proposed method is evaluated on test sets with varying Signal-to-noise Ratio (SNR) from -2 to 12 dB, while the SNR level is kept constant in each test set. In (Vincent et al., 2008) also a stacked denoising autoencoder is used to learn features from unlabeled information and consecutively limited labeled data is used to post train the encoder, making it suitable for classification purposes. In this study, in addition to the variation of levels of noise which is modeled by the addition of

Gaussian Noise from 0 to 30 dB, the amount of labeled information available for the post-training process is also taken into account and it is shown that acceptable performances from classification point of view are achievable in even extremely limited labeled information scenarios, using the proposed method. Distance learning methods, due to their intrinsic ability in dedication of regions of space to specific classes, regardless of the presence of noise in the environment, have gained significant attraction to cope with noise presence. For example, in (Zhang et al., 2019) a Siamese network employing a deep convolutional neural network as its feature extractor is used to provide reliable performance in rolling bearing fault diagnosis. In this study, presence of noise is modeled by the addition of Gaussian Noise with varying SNRs, from – 4 to 10 dB. Moreover, this study investigated the effect of data availability on the goodness of classification, by varying the amount of available information and monitoring its effect on the classification accuracy.

4. PROBLEM DEFINITION

Given X and Y as input and output spaces, respectively, we are provided with n labeled samples, $D_{train} = \{(x_i, y_i)\}_{i=1}^n$ where $y_i \in Y, x_i \in X$. We aim to construct a function f that maps input space X to the new space \mathbf{X} , $f : X \rightarrow \mathbf{X}$. To this end, we design a method that ensures that \mathbf{X} :

1. is capable of multi-level denoising.
2. can be constructed using limited available labeled samples.
3. is class discriminating.

Therefore a conventional classifier on top of the new space \mathbf{X} classifies original and noisy samples effectively.

5. THE PROPOSED METHOD

We construct a new feature space for denoising inspired by both supervised and unsupervised contrastive learning. From one hand, taking advantage of the availability of labeled samples, we employ supervised contrastive learning. As a Few-shot learning technique, supervised contrastive learning can construct a class discriminated space based on a few labeled samples. On the other hand, inspired by unsupervised contrastive learning, we augment the samples with different levels of noise. It is noteworthy that, in the case of unsupervised contrastive learning, augmentation is required due to the lack of labels for the samples. However, in this paper, we perform data augmentation for denoising purposes. Figure 1 summarizes the steps of the proposed method.

First, we augment the samples, D_{train} , with the any arbitrary levels of noise. This step is shown in the figure 1a. Therefore, considering L levels of noise, we construct $D_{train}^{noisy} = \cup_{l=1}^L D_{train}^l$ where D_{train}^l is the augmented samples with noise level l . To be consistent, let call D_{train} as $D_{train}^{original}$.

After that we construct pairs using the original and noisy samples. This step is shown in figure 1b. The possible group of pairs are:

1. (x_i, x_j) where $x_i \in D_{train}^{original}$ and $x_j \in D_{train}^{noisy}$. This group of pairs by aggregating original samples with noisy ones helps in denoising. In addition, because of pairing original samples with augmented samples with different levels noise, this group helps in multiple level denoising.
2. (x_i, x_j) where $x_i, x_j \in D_{train}^{noisy}$. This group of pairs enhances the aggregation of multiple levels of noise. Consequently, this group of pairs, when combined with the previous group of pairs, contributes to multilevel denoising. In contrast with cases where original data is only corrupted with a single level noise, known as single level denoising, multilevel denoising involves reduction of noise where data is corrupted using various levels of noise.
3. (x_i, x_j) where $x_i, x_j \in D_{train}^{original}$. By using limited labeled samples, this group of pairs is able to construct a class discriminated feature space. In fact, this group of pairs is similar to those used in supervised contrastive learning in few-shot learning (Jadon, 2020).

We calculate the similarity between paired samples (in all three groups) based on their labels. Although we augment the data, they are already labeled, which allows us to calculate the similarity between paired samples directly from their labels. Consequently, any training example takes the form of $\{(x_i, x_j), c_{i,j}\}$ where $c_{i,j}$ is the contrastive label corresponding to the pair (x_i, x_j) .

Following the preparation of the training examples (the pairs with respective contrastive labels), they are fed into a Siamese neural network shown in figure 1c; This network is constructed with two copy of feature extractor f . the network is trained using the Contrastive Loss described in equation 1. After training the network, the function f is used to map samples from input space X to the new feature space \mathbf{X} .

The last step is training a classifier, utilizing the feature space derived by the network, \mathbf{X} . In this study, K-nearest neighbor (KNN) is used to carry out the classification step. We consider KNN as the best choice of classification model in this study, mainly due to its pure distance-based mechanism, which makes it a great metric to evaluate the effectiveness of a feature space, in providing sufficient separability of classes.

6. EXPERIMENTS

In this study, the effectiveness of the proposed method is evaluated using the dataset provided by the Southeast University (Shao, McAleer, Yan, & Baldi, 2018). The referenced dataset consists of both bearing and gearbox signals, however, in this study we only utilized the bearing dataset. Five different health classes are taken into account for bearings,

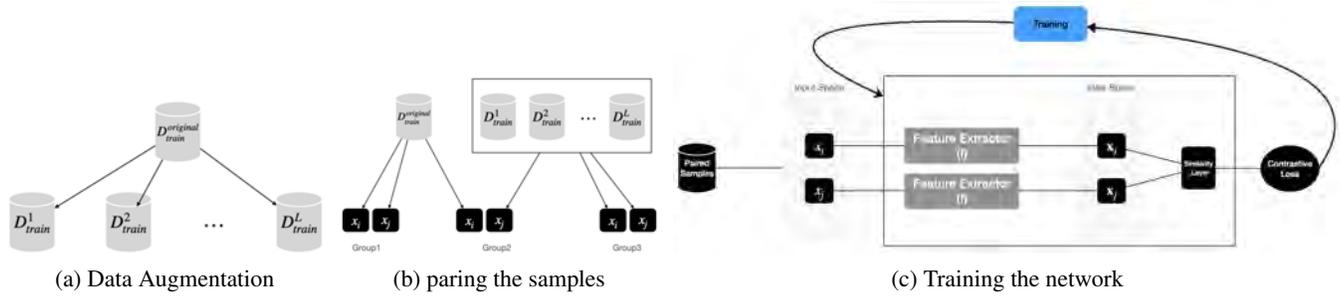


Figure 1. The three steps of the proposed method

including healthy operation, inner ring fault, outer ring fault, rotating ball fault and combination of inner and outer ring fault. Moreover, two rotational speeds, as loading conditions, are included in this dataset (20 and 30 Hz). Various channels of accelerations coming from various locations of the test bench are provided in this dataset and we used the signals provided by the second channel in this study. The original time series identical to each load-fault combinations are split to 1024 point long signals in time domain. Consecutively, Fast Fourier Transform is used to derive the frequency domain observations from time domain observations, as bearing faults are significantly easier to diagnose in frequency domain. Employment of FFT on the time domain signals would provide us with 512 point long frequency domain signal.

For the experiments, we augment the data by adding Gaussian noise with two different Signal-to-noise ratios (SNR), -2 and -4 dB. Mathematical definition of SNR can be seen in 2, where P_{signal} and P_{noise} are powers of original signal and noise, respectively. It is worth noting that any arbitrary noise can be added to data. In addition, all of the provided results are the average of five runs.

$$SNR_{db} = 10 \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right) \quad (2)$$

The Siamese network used in these experiments utilizes a three-layered multi-layered perceptron as the base feature extractor. Hyperbolic tangent is used as the activation function for all the neurons in the network; moreover, number of neurons per layer are as 512-256-128. ADAM optimizer with the learning rate equal to 0.0001 was used to train the network and 1000 iterations provided satisfactory training process in all of the experiments.

As a demonstration that our proposed method can construct a noise-robust feature space, we show that there is no distinction between original and noisy samples (regardless of the noise level) in the new space.

To conduct experiments that demonstrate our method’s ability to work with limited data, first of all, we separate 40 percent of all samples for use as a test dataset, in order to have the

same test data for all experiments. We construct the training datasets using the remaining 60 percent of samples. By varying the number of training examples, we conduct three experiments. For the first one, we use all 60% of the remaining samples as the training dataset. The second, third training datasets consist of 20%, and 1% of the 60% remaining samples, respectively.

In addition, we aim to construct a model that is robust to the different levels of noise (strong noise levels) that may occur in an environment where the model will be used; to simulate such a condition, we add noise to the test data and evaluate the performance of our method and provide the results for the four following situations.

1. original test samples. These results are shown in the first set of bars in each sub-figure named by *Result on No noise*.
2. The corrupted test samples with -2 dB noise. These results are shown in the second set of bars in each sub-figure named by *Result on -2 dB*.
3. The corrupted test samples with -4 dB noise. These results are shown in the third set of bars in each sub-figure named by *Result on -4 dB*.
4. The combination of original and corrupted samples with -2 and -4 dB noise. These results are shown in the last set of bars in each sub-figure named by *Result on overall*.

Furthermore, it is worth mentioning that our method, regardless of how many levels of noise are taken into consideration, works by unifying the original and noisy data (augmented data by noise); so that in the constructed space, they cannot be distinguished (In this way the effect of noise will be removed). To demonstrate that the original and noisy data are adequately unified by our method, for all experiments, we first extract a new feature representation, then on the top of the constructed space, we train KNN (with $k = 5$) using

1. original samples (The red bars in the plots that is named with *train KNN on No-Noise*),
2. samples corrupted with -2 dB noise (The blue bars in the plots that is named with *train KNN on -2 dB*),

3. samples corrupted with -4 dB noise (The purple bars in the plots that is named with *train KNN on -4 dB*),
4. all original and noisy samples (The gray bars in the plots that is named with *train KNN on Overall*).

As we can see in the bottom sub-figures in figures 2 to 4, in all cases, the results of KNN with different training sets are the same; this means the training data for KNN (No-noise, -2 dB, -4 dB, and overall) are aligned to each other in the constructed feature space. Thus, the same result from, for example, KNN on 'No-Noise' (red) and train with 'all noises' (grey) is proof that our proposed method is functioning and the constructed space is robust to noise with varying levels. In addition, the fact that such results have been obtained even after decreasing the number of training examples indicates that our method is capable of denoising data even when the number of training data is insufficient, although the performance in terms of the detection accuracy is affected by this factor.

In order to emphasize the effect of the augmentation in our method, we remove this step and redo the experiments. The first sub-figures (sub-figures on the top) in figures 2 to 4 show the results. In fact, we can interpret the first sub-figures as the results of applying contrastive learning to data; we just pair the available labeled samples and train the Siamese neural network with contrastive loss. Comparing every two sub-figures demonstrates how data augmentation part of the proposed method is crucial for denoising; As we can see in the first sub-figure of figures 2 to 4, when we do not perform the augmentation process, training KNN on different levels of noise provides different results regardless of the test case. In fact, these differences indicates that the constructed space which is obtained without data augmentation is not robust to the noise.

Moreover, we conduct another experiment and compare the results of our method with KNN and denoising autoencoder. The results are shown in the figure 5. In fact, we compare the results of our method with the results of KNN applied to the original samples. The reason we perform this comparison is because we applied KNN to the constructed feature space as well. Therefore, this comparison illustrates the capabilities of the constructed feature space.

In addition, we compare the results of the proposed method with those of the conventional denoising autoencoder. We designed this comparison to demonstrate the superiority of our methods in removing multiple noise levels with insufficient available samples. Our approach is to train a denoising autoencoder, using the -2 dB, -4 dB and original training sets as the input and corresponding noise-free version of training sets as the output. On top of the constructed space with the denoising autoencoder, we train a KNN classifier with $k = 5$. As with other experiments, we use the same 40% of the samples as a test dataset. We consider 5% of the remaining samples as labeled datasets.

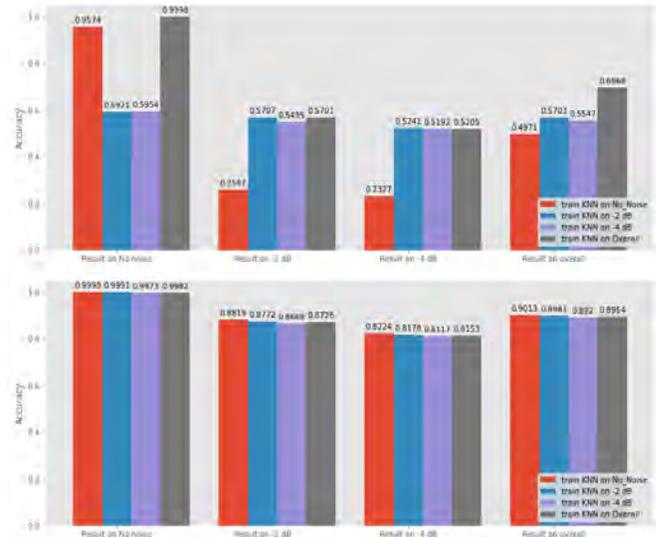


Figure 2. All training samples are used in this experiment. The first sub-figure is the results of eliminating data-augmentation. The second sub-figure shows the results of our proposed method.

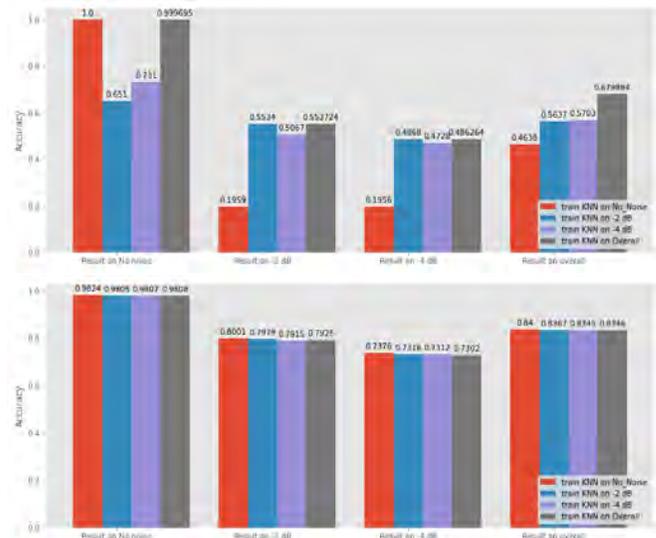


Figure 3. 20 Percent of training samples are used in this experiment. The first sub-figure is the results of eliminating data-augmentation. The second sub-figure shows the results of our proposed method.

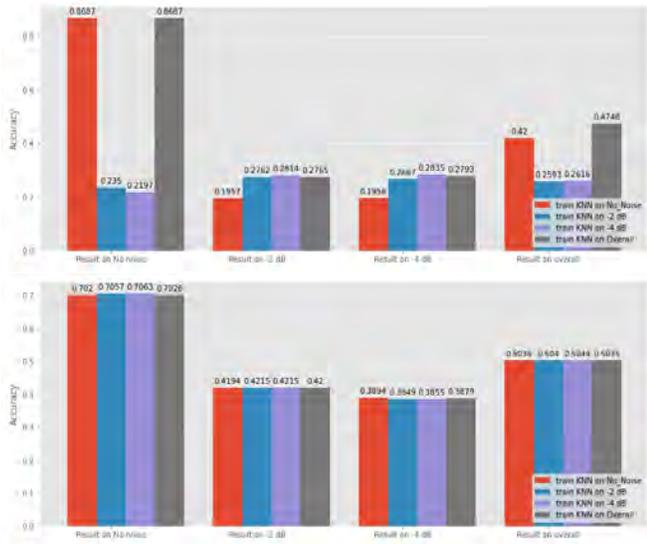


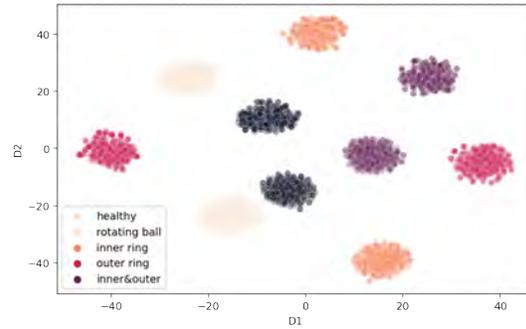
Figure 4. 1 Percent of training samples are used in this experiment. The first sub-figure is the results of eliminating data-augmentation. The second sub-figure shows the results of our proposed method.

Similar to the previous experiment, we evaluate the methods on the original test samples, corrupted test samples with -2 and -4 dB noises and the combination of original and corrupted ones. As we can see in figure 5, in the noisy environments, our method outperforms other.

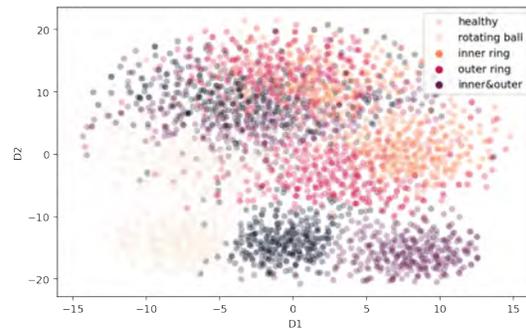


Figure 5. Evaluation of our method in comparison with others; The results are obtained using 5 percent of the samples

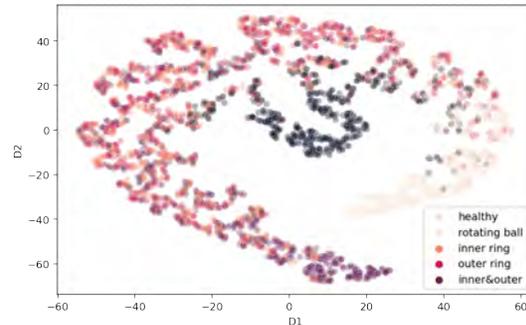
Moreover, we use t-SNE to visualize the effects of our proposed method. Figure 6a illustrates the training samples. As can be seen, this dataset is not noisy. However, due to the fact that we use data from two different loads, there are two distinct clusters per health class. In Figure 6b, we see the test samples that have been corrupted by -2dB noise. Considering these two figures, we are able to conclude that a model trained with clean training samples will not be robust to noise and therefore will experience performance degradation. To address this problem, we aim to reduce the effect of noise, in a new representation space. As a critical part of our proposed method is data augmentation, we show the constructed



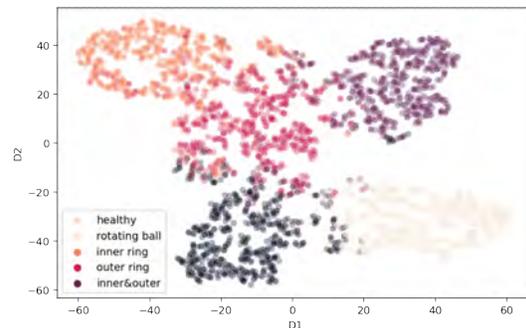
(a) Training samples



(b) Corrupted test sample. The samples are corrupted by -2 dB noise.



(c) Test samples in the new constructed feature representation when the augmentation step is ignored.



(d) Test sample in the new constructed feature representation when the data augmentation step is performed.

Figure 6. t-SNE is used to visualize training and test samples in the original and constructed space, and to compare the constructed space with and without the data augmentation step.

space with and without this step. Figure 6c shows the test samples in the constructed space when data augmentation is ignored. We can see that samples from different health states are overlapped. Figure 6d demonstrates the test samples in the constructed new representation when data augmentation is employed. As can be seen, our method can reduce noise as well as gather samples from different loads, resulting in a higher degree of accuracy. This visualization is related to the experiments for which all training samples are used to construct a new space, whose results can be found in Figure 2.

7. CONCLUSION

In this paper, we propose a feature representation learning method for the purpose of denoising. It is possible to remove multiple levels of noise through this technique, even when enough labeled samples are not available. To achieve this goal, we augment the samples with different levels of noise, inspired by unsupervised contrastive learning techniques. Using the original samples and the corrupted ones, we pair the samples. Then, using the prepared paired samples, we train a Siamese neural network with Contrastive Loss function. Training the network results in a feature extractor that maps the samples to a new space. In this space, the corrupted samples are aggregated with the original samples, resulting in denoising. Moreover, since the new space is constructed using contrastive learning, not only are the classes separated but also the new space can be achieved by using a small number of labeled samples. With the SEU dataset, we conduct several experiments with different amounts of labeled samples. The effects of denoising can be observed in each case. We also compare our method to the results of the denoising autoencoder in the absence of sufficient labeled data.

8. ACKNOWLEDGEMENTS

This research has been funded in part by the Knowledge Foundation and by Vinnova, Strategic Vehicle Research and Innovation programme.

REFERENCES

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020).

A simple framework for contrastive learning of visual representations. In *International conference on machine learning* (pp. 1597–1607).

- Chicco, D. (2021). Siamese neural networks: An overview. *Artificial Neural Networks*, 73–94.
- Jadon, S. (2020). An overview of deep learning architectures in few-shot learning domain. *arXiv preprint arXiv:2008.06365*.
- Le-Khac, P. H., Healy, G., & Smeaton, A. F. (2020). Contrastive representation learning: A framework and review. *IEEE Access*, 8, 193907–193934.
- Liu, X., Zhou, Q., Zhao, J., Shen, H., & Xiong, X. (2019). Fault diagnosis of rotating machinery under noisy environment conditions based on a 1-d convolutional autoencoder and 1-d convolutional neural network. *Sensors*, 19(4), 972.
- Oh Song, H., Xiang, Y., Jegelka, S., & Savarese, S. (2016). Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4004–4012).
- Shao, S., McAleer, S., Yan, R., & Baldi, P. (2018). Highly accurate machine fault diagnosis using deep transfer learning. *IEEE Transactions on Industrial Informatics*, 15(4), 2446–2455.
- Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on machine learning* (pp. 1096–1103).
- Wang, S., Wang, D., Kong, D., Wang, J., Li, W., & Zhou, S. (2020). Few-shot rolling bearing fault diagnosis with metric-based meta learning. *Sensors*, 20(22), 6437.
- Zhang, A., Li, S., Cui, Y., Yang, W., Dong, R., & Hu, J. (2019). Limited data rolling bearing fault diagnosis with few-shot learning. *IEEE Access*, 7, 110895–110904.
- Zhao, W., Lu, C., Ma, J., & Wang, Z. (2015). A deep learning method using sda combined with dropout for bearing fault diagnosis. *Vibroengineering Procedia*, 5, 151–156.

Automating Critical Surface Identification and Damage Detection Using Deep Learning and Perspective Projection Methods

Gautam Kumar Vadisala¹, Anurag Singh Rawat², Abhishek Dubey³, Gareth Yen Ket Chin⁴ and Fabio Abreu⁵

^{1,2,3}*Schlumberger, Building 8, Office 301, Commerce zone IT Park, Pune, Maharashtra-411006, India*

GVadisala@slb.com

ARawat4@slb.com

ADubey4@slb.com

^{4,5}*Schlumberger Technology Corporation, Rosharon Testing and Subsea Center, Rosharon, Texas-77583, US*

GSchin@slb.com

FAreu3@slb.com

ABSTRACT

With an increased collection of data, assessing the health of an asset and designing recommendations or executing response actions via prognostics and health management (PHM) has made great advances. These actions can be corrective or preventive depending upon the risk of failure or the cost of repair. As downhole testing tools operate in extreme environments, they are subjected to conditions like elevated temperature, shocks, vibrations, and pressures. The dump mandrels used in the process are prone to wear and tear like scratches, pits, and corrosion, which may cause operational failure. If these damages and their degree goes undetected and no remedial actions are taken, possibilities of non-productive time (NPT) and financial losses increase drastically. This paper aims to develop a fitness inspector which uses Computer Vision and Deep Learning to identify critical surfaces of these tools and the damage within them. This will help the Subject Matter Experts (SMEs) by replacing the qualified workforce provided by them and reducing the time consumed to gauge the health status of all the tools as the diagnosis can be made in real-time.

1. INTRODUCTION

Health management is an important aspect of tool lifecycle management. With correct management of data, we can have full visibility into the health of an asset throughout its lifecycle, from design to production to obsolescence. We have access to the past data of the whole fleet to analyze any anomaly and diagnose why it happened. By connecting the

Gautam Kumar Vadisala et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

learnings from past data to the real-time data coming from an asset, we can get a current health assessment and diagnose the root cause of an anomaly as it is happening.

After operating in fields, Down Hole Testing tools such as dump mandrels have damage on the sealing surfaces and O-ring grooves. This mandates a tool inspection by an SME after every use to decide whether it would be fit for the next job. This process invites two types of risks which are, scrapping good tools too early and using a questionable tool in a million-dollar worth of field operation. Also, this would require a lot of effort by subject matter experts (SMEs) in terms of the inspection time. We aim to reduce this inspection time and provide a data-based decision-making process.

We aim to propose a real-time visual platform during an inspection, that would take a video of the tool being inspected as an input and provide the decision of whether the tool can be used for the next operation or to be sent for repair. This video can be taken from a Digital single-lens reflex (DSLR) camera or a mobile phone in a fixed setup. Although, we felt that the former would provide better results (this is explained in the further sections of the paper).

As dump mandrels are cylindrical, we need to record the video of the mandrel by rotating it along its axis to capture all its surface area for inspection. The video is then further processed into individual frames for analysis. In each frame, we first detect a reference point through which we locate the critical surfaces of the tool and then identify the damage on those surfaces. We also need the engineering designs of the mandrel to locate the critical surfaces or the area of interest in the frames. We perform these steps on all frames and produce the final result.

We consider the center of the mandrel edge as a reference point as the critical surface areas are mentioned from this edge in the engineering designs. To detect this point, we compute the mandrel boundaries in each frame by a Deep Learning based Object Detection method. After locating the edge of the mandrel in the video frame, we project the critical surfaces on it by estimating the distance of the critical surface in terms of pixels. We consider the diameter of the tool as a reference measure throughout the video and use it to convert the actual distance in inches (or mm) to pixels. As every prediction result from the model is probabilistic, this causes dislocation of critical surfaces from their actual location, sometimes by a large degree in the resultant video if the input video was shaky and the rotational speed of the video was not constant. Another major issue we encountered was the conical projections of the tool in the video. As the mandrel is cylindrical, it is difficult to precisely follow the surface at different points by this method. So, this deep-learning-based method is not able to address this issue as we can only project vertical lines for critical surfaces given their distance from the reference point.

To solve these issues and to make the projection of critical surfaces similar throughout the video, we went for a fixed apparatus where the camera would be placed at a fixed distance and the tool would be rotated at a constant rotational speed. We can then leverage Camera Projection methods to convert the real-life distances in inches (or mm) to the distance in pixels. This would also take care of the curvilinear surfaces as our projections follow the 3D nature of the tool in the video.

For damage detection on the tool surface, we train a Deep Learning based Semantic Segmentation model based on U-net architecture. As the surface damages would be small and different, image-processing-based semantic segmentation methods are not helpful. Also, this model only requires a few images for a good-enough solution and can be improved with more data. After detection of damage, we project the damage identified on the critical surfaces or our area of interest.

After the projection of damage on the critical surfaces, we can leave it to the SME to decide on further actions or we can create a metric to compute the percentage of damaged pixels from the whole critical surface areas in pixels. By using this methodology, we can provide an automated solution for end-to-end tool inspection. Our method can be easily replicated for other tools with known geometry to have a surface-level inspection.

2. RELATED WORK

Cylindrical objects like Mandrels tend to have perspective effects when captured in an image. This means the points on the cylindrical surface which are at some distance from the straight line of vision of the camera will appear in curves on the image. For example, points on the left side of the camera will appear curved and their focus will be on the camera

center. A solution for this problem of cylindrical objects producing perspective effects like conical sections in images was previously presented in the article (Berveglieri & Tommaselli, 2018). Their method aims at performing a continuous reconstruction of 3D cylindrical patches with high accuracy. They collect the images at different viewpoints and then fit the corresponding image patches using a modified geometric transformation for Least Square Matching (LSM) (Gruen, 1996). The image acquisition is performed by displacing the camera in a line path parallel to the cylindrical axis, as shown in the below figure (Figure 1).

In Figure 1, three views of a strip over the cylinder that correspond to the strip over the cylinder are displayed. The strip appears as a horizontal shape (the main axis appears as a straight line) when the image is collected from a normal view (the projecting ray to the strip over the cylinder). If the projecting ray of the strip path is oblique, then the strip has a curved shape. This occurs due to the cylindrical shape of the object and the change in camera viewpoint. Given two image patches or regions that refer to the same area over the cylindrical diameter but with different perspective views, parallaxes will occur due to depth and orientation variations.

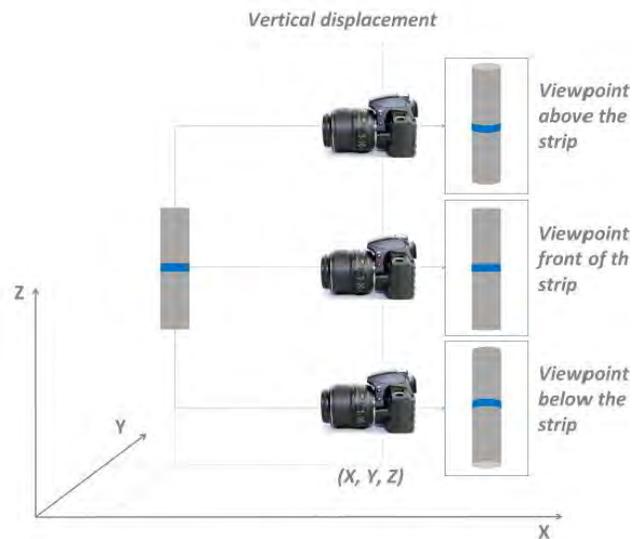


Figure 1. Image acquisition using camera displacement (Berveglieri & Tommaselli, 2018)

The method proposed by Berveglieri and Tommaselli (2018) aims to reconstruct the 3D cylindrical surface by using a geometric transformation T (as mentioned in Figure 2) with further refinement by an adaptive least square matching (ALSM) (Gruen, 1985), to accurately map a point from an image $I_1(x, y)$ to its respective correspondence in an image $I_2(x, y)$ with sub-pixel precision. Although the above method provides a solution to the conical sections in images, we are looking for a solution without the surface reconstruction. Also, this method brings up the data requirement of collecting the videos of a mandrel across different viewpoints. So, we

propose a method where we use Camera Projection techniques and project a real-world 3D object onto a 2D image. Also, our solution only requires one video of the mandrel.

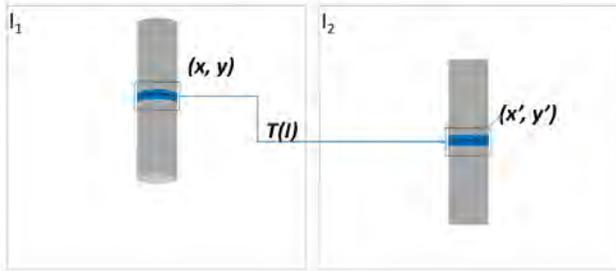


Figure 2. Matching using Least Squares (Berveglieri & Tommaselli, 2018)

3. METHODOLOGY

The proposed solution has three stepped approaches. First, the reference point is identified in the video frame. Second, the area of interest i.e., the critical surface areas which are of concern are identified in the same frame. Then the damage is detected within the critical surface on the mandrel in that frame. For final decision making, we combine the above solution for all the video frames and consider the damage within the critical surfaces.

3.1. Reference Point Identification

To identify critical surfaces on a mandrel, we first need a reference point from which we can project them on the mandrel in the image. Since the dump mandrel is cylindrical, we felt the center of the edge would be suitable for this purpose. But to find this point, we need to detect the mandrel boundary in the image. There are many methods from which we can estimate the mandrel boundaries, but from our observations, edge detection methods give approximate results for computing the object boundary.

3.1.1. Edge Detection

The above time constraint prompted us to look for a solution that would be easy to compute and in a lesser amount of time. We felt that if the mandrel is in the foreground and the video is recorded with a clear background and without much distortion, we can employ edge detection methods such as Canny Edge Detection (Canny, 1986) to estimate the object boundary in the image. This edge detection algorithm applies Gaussian smoothing and computes the intensity gradients to detect a wide range of edges in images. We can choose a threshold to filter out the edges that would be useful for the object of our interest. An example of Canny Edge detection is shown in Figure 3 where we could detect the edges of the objects present in the image.

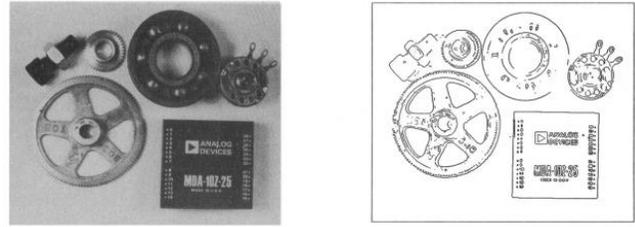


Figure 3: Canny Edge Detection Example (Canny, 1986)

By using this method, we can compute the reference point coordinates within a fraction of a second. Although it requires a specific setup while recording the video, the results from this method are accurate and close to the actual object boundaries. So, we preferred this method to compute the coordinates of the reference point.

3.2. Critical Surface Identification

Critical surface areas on the mandrels refer to the sealing surfaces (point of contact) for the O-rings. Any damage and defects in these areas result in oil spilling and hence the mandrels should be reused, repaired, or discarded based upon the degree of the damage. To identify and project the critical surfaces (O-ring grooves) in the frames of the video, the previously mentioned fixed setup is used where the camera remains stationary. The concepts of camera projection (Collins, 2007) are then used to convert the real-life coordinates of the grooves to pixel coordinates in the image. The projection equation shown in Eq. (1) is used to achieve this.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \underbrace{\begin{bmatrix} -f/s_x & 0 & +o_x & 0 \\ 0 & -f/s_y & +o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{M_{int}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{M_{ext}} \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix} \quad (1)$$

Here, U, V, and W are the real-world coordinates that need to be converted into the pixel coordinates (u, v) of the image. Variables s_x and s_y are the pixel size of the camera sensor along the x and y axes respectively, further explained in Eq. (13) and Eq. (14). The focal length of the camera is denoted by f and the camera's film plane center offsets from its sensor's pixel array origin along the x and y axes as explained in Eq. (9) and Figure 8 are denoted by o_x and o_y .

3.2.1. Extrinsic Parameters

The real-world coordinates are first converted into camera coordinates using extrinsic parameters R in Eq. (2) and T in Eq. (3) which together form M_{ext} in Eq. (1) and enable this transformation.

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2)$$

$$T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (3)$$

R is the measure of rotation required to align the real-world and camera coordinate axes. This means that (r_{11}, r_{12}, r_{13}) is rotational units needed in the camera x-axis, (r_{21}, r_{22}, r_{23}) in the camera y-axis, and (r_{31}, r_{32}, r_{33}) in the camera z-axis. This can be better understood by looking at Figure 4. As you can see in this figure, the train camera’s x-axis resonates with the real-world z-axis, therefore (r_{11}, r_{12}, r_{13}) becomes $(0, 0, 1)$. The y camera axis resonates with the real-world x-axis but is in opposite directions, hence (r_{21}, r_{22}, r_{23}) becomes $(0, -1, 0)$. The Z camera axis resonates with the real-world x-axis, therefore (r_{31}, r_{32}, r_{33}) becomes $(1, 0, 0)$. This way we get R_{train} mentioned in Figure 4.

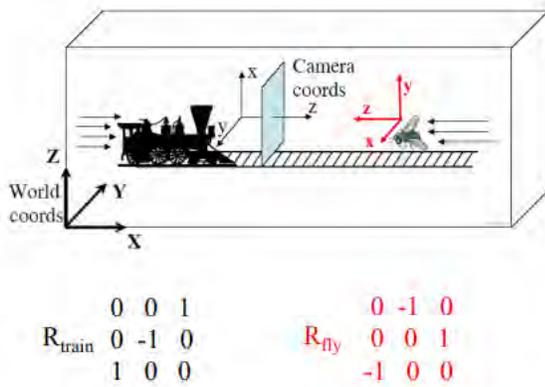


Figure 4. Extrinsic Parameter – Rotation (Collins, 2007)

T, on the other hand, is the location of the camera relative to the real-world coordinate system. t_x, t_y, t_z are, therefore, the camera’s position on world coordinates’ x, y, and z-axis respectively. They are a measure of translation/movement that the camera axis needs to undergo to be aligned to the real-world axis as shown in Figure 5.

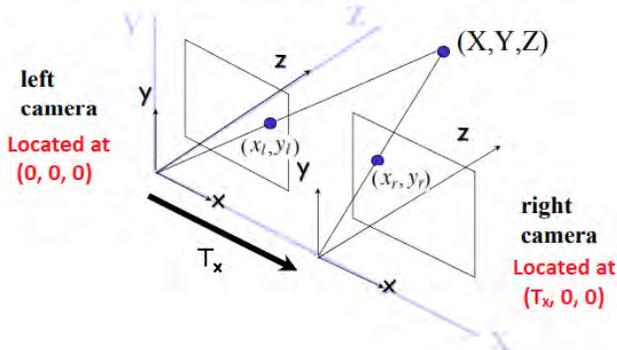


Figure 5. Extrinsic Parameter - Translation (Collins, 2007)

3.2.2. Perspective Matrix Equation

The concept of perspective projection helps us to understand how 3D coordinates can be projected to a 2D plane. This is then employed to convert the 3D camera coordinates which we calculate using extrinsic parameters, to 2D film coordinates. The film plane of the camera is located at f (focal length) units along with the optic (Z) axis of the camera coordinate system (see Figure 6).

The perspective projection equations i.e., Eq. (4) and Eq. (5) can be derived from the rule of the similar triangles (see Figure 6 and Figure 7)

$$x = \frac{fX}{Z} \quad (4)$$

$$y = \frac{fY}{Z} \quad (5)$$

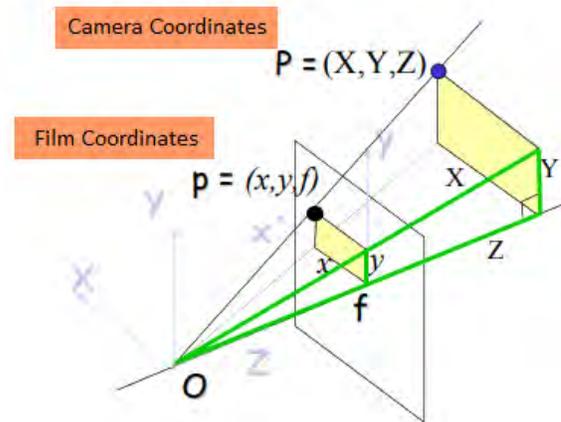


Figure 6. Projection on Film Plane (Collins, 2007)

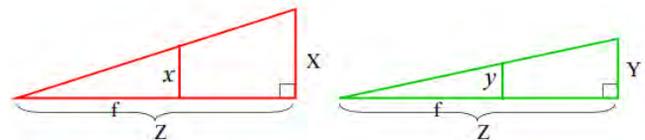


Figure 7. Object and projection on film plane forming similar triangles (Collins, 2007)

The perspective projection equations can also be represented as a matrix by introducing homogenous coordinates. Homogenous coordinates represent a 2D point (x, y) by a 3D point (x', y', z') , by adding a fictitious third coordinate. Given (x', y', z') , the 2D point can be recovered as shown in Eq. (6) and Eq (7).

$$x = \frac{x'}{z'} \quad (6)$$

$$y = \frac{y'}{z'} \quad (7)$$

This way, we transform the perspective projection equation (Eq. 4 and Eq. 5) into a matrix (Eq. 8).

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (8)$$

3.2.3. Intrinsic Parameters

The images projected on the film plane are further digitized, which poses the need for us to transform the film coordinates of interest derived from the perspective projection matrix into pixel arrays.

This is achieved by the usage of intrinsic parameters O and S and shown in Eq. (9) and Eq. (10) respectively.

$$O = [O_x, O_y] \quad (9)$$

$$S = [S_x, S_y] \quad (10)$$

O is the offset or the image center in the film plane that needs to be added to the derived film coordinates, as the film and pixel coordinate systems along with their origins are different (seen Figure 8).

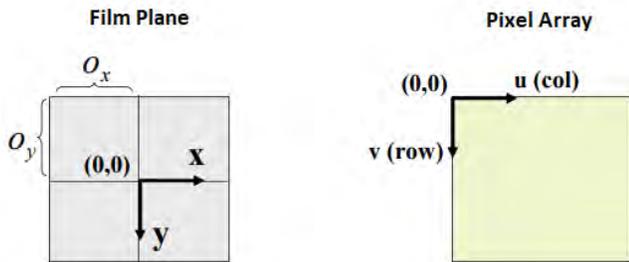


Figure 8. Offsets O_x and O_y (Collins, 2007)

With this, we can calculate the pixel coordinates (u, v) as per Eq. (11) and Eq. (12).

$$u = x + o_x \quad (11)$$

$$v = y + o_y \quad (12)$$

Conversely, we can calculate o_x and o_y if we find the pixel and film coordinates for any one point.

The second intrinsic parameter i.e., S represents the pixel size s_x and s_y . s_x and s_y give us a measure of how many units of distance (mm, inch, etc.) a pixel covers on the image plane (camera sensor). It can easily be derived as in Eq. (13) and Eq. (14).

$$s_x = \frac{\text{sensor width}}{\text{image width in pixels}} \quad (13)$$

$$s_y = \frac{\text{sensor height}}{\text{image height in pixels}} \quad (14)$$

The pixel size is incorporated into Eq. (11) and Eq. (12) effectively changing them to Eq. (15) and Eq. (16)

$$u = \frac{x}{s_x} + o_x \quad (15)$$

$$v = \frac{y}{s_y} + o_y \quad (16)$$

3.2.4. Projecting Critical Surface Grooves

Since the mandrel being cylindrical, is a circle in the real-world $Y-Z$ plane, we can get the real-world Y and Z coordinates of the grooves (red bands in Figure 9) along the circular surface with the help of the mandrel's radius (as shown in Figure 9).

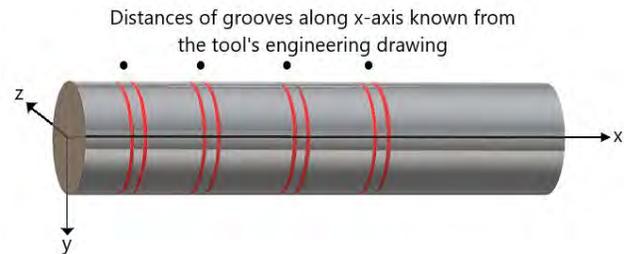


Figure 9. Figure showing a Mandrel and the critical surface grooves

The X coordinates of the grooves are taken from the engineering design of the tool. These real-world (U, V, W) coordinates are used along with the extrinsic and intrinsic parameters as shown in Figure 10 to get the pixel coordinates (u, v) of these grooves in the images.

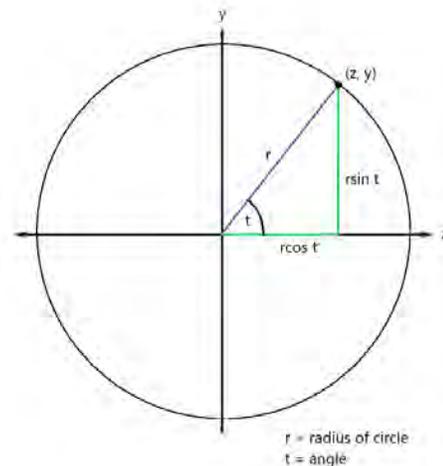


Figure 10. Offsets O_x and O_y

3.3. Damage Detection

To detect the damage on the mandrel in the image, we use a computer vision technique called Semantic Segmentation in conjunction with Deep Learning. We train a model based on U-Net architecture to predict which pixels in each image were damaged.

3.3.1. Semantic segmentation

Semantic segmentation is a process where every pixel in an image is associated with a certain label or class. It helps us to distinguish between predefined multiple categories in an image with pixel-level granularity (as in Figure 11).

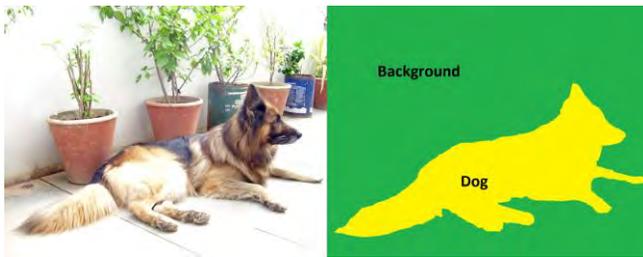


Figure 11. Semantic Segmentation predicting 2 classes (Dog and Background)

3.3.2. U-Net

We use U-Net architecture (Ronneberger, Fischer and Brox, 2015) to employ Deep Learning and achieve Semantic Segmentation. U-Net has 2 logical components, an encoder, and a decoder. The encoder, which can be any Convolutional Neural Network (CNN) architecture-based feature extractor, extracts feature maps from images at a resolution and downsamples the images before repeating the same process. This way we get feature maps of an image at different resolutions. The decoder takes the feature maps of the lowest resolution and upsamples them. It then takes the upsampled feature maps and merges them with the feature maps that were extracted at that resolution earlier by the encoder. This process is repeated by the decoder till we get the final output with the original resolution of the input images. This output is the representation of the various categories present in the images and the pixels belonging to them.

4. DATA AND EXPERIMENTS

4.1. Reference Point Identification and Critical Surface Detection

We have created a fixed setup to rotate the mandrel and capture its surface area. The specifications of the camera are mentioned in Table 1. To capture all the surface area, the mandrel would be rotated along its axis at a fixed speed, so it would not distort the mandrel recording in the video.

Table 1. Camera Specifications

Specification	Value
Camera Used	Nikon D550
Sensor Size	23.5 mm x 15.60 mm
Pixel Size	0.0039 mm (For 24 MP Image)
Focal Length	18 mm

As shown in Figure 12, we fixed the camera on a stationary point within a certain distance to capture the surface area but not too far which would make the damage on the surface invisible to the naked eye. As mentioned before, the reference point can be easily detected from the mandrel boundaries in the image given the clear background. Based on our observations, The Camera Projection method can locate the critical surface area with a **maximum error of 2.54 mm** (0.1 inches) as per our experiments with our setup. If we expand the critical surface areas by 2.54 mm on either side, we can include all the actual critical surface areas in the image that would otherwise be missed due to this possible error. The final projections can be seen in the below figure (Figure 13) where the area enclosed between the white curves on the mandrel surface is our critical surface. The small black markings on the surface represent the markings of the actual critical surface. As we can see, our projections are closer to the actual areas and follow the conical nature of these curves along the surface.

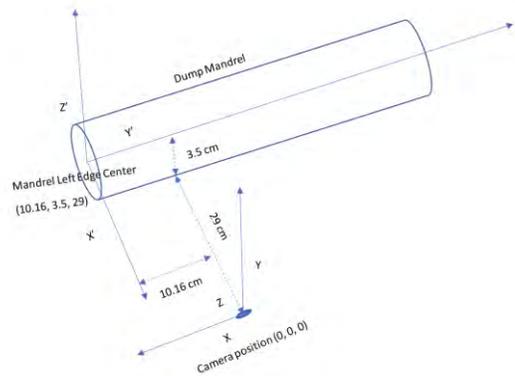


Figure 12. Camera Setup for Critical Surface Identification

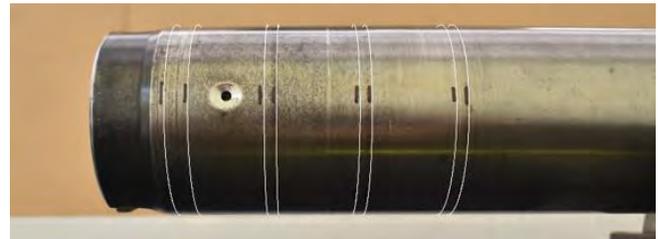


Figure 13. Critical Surface Projection

As we used a uniform background in our experiments (visible in Figure 13), the foreground (dump mandrel) stands out, and

the edge detection method could be used to compute the boundary of the mandrel in the image, which helps us to detect a reference point and project critical surfaces. If the background is noisy, the edge detection method alone won't be useful as the edges from any texture or object in the background can cause interference, and usage of deep learning based methods would be required to extract the boundary of the mandrel.

4.2. Damage Identification

In our first model training, we recorded 20 videos of the damaged mandrels for the damage detection model training. We converted these videos into frames and annotated them where the damage is visible with our naked eye. For semantic segmentation, we need to annotate on pixel level and assign them a class. For implementation purposes, we have combined all the damages that can occur on the mandrel into one class. So, we will be predicting two classes i.e., damage and background (no damage), which is a binary classification at pixel level and outputs the probability of each image pixel belonging to the two mentioned classes. A segmentation mask would be prepared to mark damage in the image, which is a 2D array of 1s and 0s where 1 indicates damage on the (x, y) coordinate in the image and 0 indicates background. The frames and their corresponding segmentation masks are used as input and output for training the semantic segmentation model.

The model takes a 4D array as input i.e., n images with RGB (Red, Green, and Blue) values, and returns a 4D array as output. The array returned as output would have, for every input image, the probability of the pixel being damaged as well as the probability of it being the background. Comparing these probabilities for every pixel of all the images, we get a mask (2D array) for every input image that tells us the image coordinates of damages and background (no damage). We used the previously discussed U-net architecture by Ronneberger et al. (2015) with ResNet (He, Zhang, Ren and Sun, 2016) and EfficientNet (Tan & Le, 2020) backbones. The EfficientNet-based U-net model fared better than the ResNet-based one for damage detection.

The model was trained by using EfficientNet architecture (Tan & Le, 2020) as the encoder for the U-net model (Ronneberger et al., 2015). We also employed transfer learning as the weights from a model pre-trained on ImageNet (Deng, Dong, Socher, Li, Li and Fei-Fei, 2009) were used. The dataset, which consisted of 300 images extracted from 20 videos of the tools rotating on their axis was split into train, test, and validation sets in a ratio of 70:15:15. The model was then trained by defining the decoder which could upsample features as per the U-net architecture, using Adam (Kingma & Ba, 2015) as the optimizer algorithm and Binary Crossentropy as our loss function.

Also, one whole image of the tool provided to the model as input would be compressed way too much horizontally (see Figure 14). This would, in turn, distort the dimensions of the damages as well and affect the model's capability to detect them. To overcome this, we decided to split any given input image into multiple slices (Figure 15) and send all of them as a batch for damage identification which can later be stitched back together after the model processes them into one whole output image.

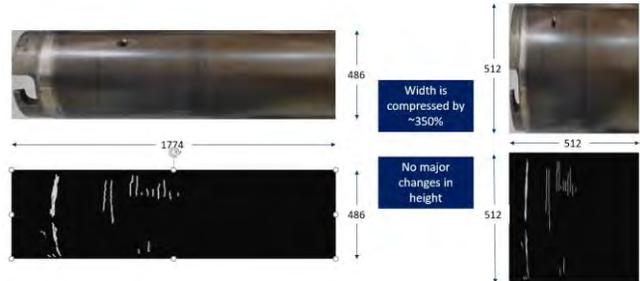


Figure 14. Original image and its damage representation (left) being compressed as the model input (right)

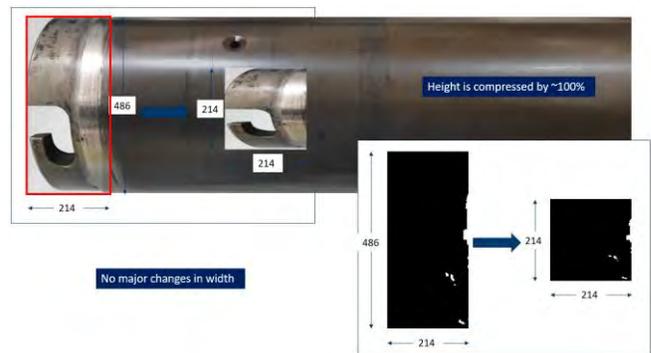


Figure 15. A slice of the input image (enclosed in red) and its damage representation before and after compression.

In general, average pixel accuracy across all classes is used as an evaluation metric for Semantic Segmentation models. As the occurrence of damage in the images is very low, the pixel accuracy metric did not help differentiate between good and bad models. So, we chose Dice Coefficient as our evaluation metric for the model. As mentioned in Eq. (17), the dice coefficient for input sample A and output sample B is defined as twice the intersection of the pixel area divided by the sum of the individual areas of the samples.

$$\text{Dice coefficient} = \frac{2(A \cap B)}{|A| + |B|} \quad (17)$$

The dice coefficient describes the percentage of overlap we can expect from the predicted pixels compared to ground truth pixels. With the above training, we observed a **mean dice coefficient of 0.7** which is good for locating the damage in the images. This means that SMEs can expect at least 70% of the damage to be detected in the critical surface areas. As the damage areas are very scarce and small in the tools and

training data, we expect we can increase the damage detection percentage with more data to train the model. As the model outputs a probability of damage on each pixel, we can apply a probability threshold to select the damages on the image. Although it is not clear in our use case, we noticed that the mean dice coefficient seems to be slowly decreasing and drops to a lower value when plotted against the probability thresholds. The huge drop in the mean dice coefficient values occurred for threshold values greater than 0.5. So, we chose a threshold where the mean dice coefficient is steady before dropping to a lower value, which provides good results when compared to the lower threshold values.

During experimentation, it was also noticed how lighting conditions could prove to impact damage detection adversely. As you can see in Figure 16, given the lighting conditions and the reflective surface of the tool, the features of the surface are hard to make out and any damage in that area would be unidentifiable.



Figure 16. Reflections on tool due to unfavorable lighting conditions

4.3. Final Results

As our area of interest is the critical surfaces on the mandrel, we need the damage that was in these areas. To obtain that, we would require two image masks. First is a 2D image mask which contains a mask of critical surface areas on the mandrel. The second one is the output mask obtained from the above damage detection model. Then, we combine the above two masks by applying bitwise AND on the two image masks. This application filters out the pixels which have a value of 1 on both masks. The result can be seen in the below figure (Figure 17) where the red pixels show the damage on the surface and the white curves enclose the critical surface area on the mandrel.

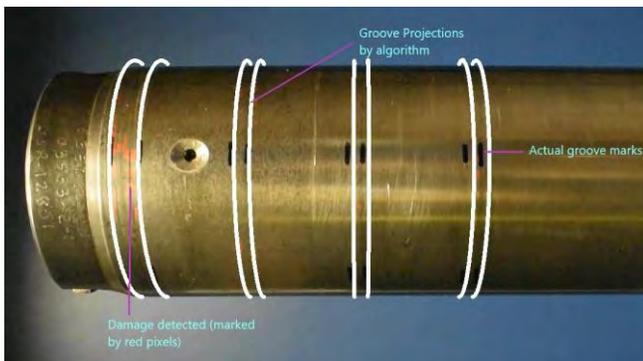


Figure 17. Damage within Critical Surface Area

The software for the experiments mentioned in this paper is developed in Python using OpenCV (Bradski, 2000) library for image processing. We have used an NVIDIA Tesla K80 GPU with 12 GB RAM and a CPU of 16 GB RAM for training and inference of the U-net model. Using this hardware, we were able to compute the inference of an image in 0.2s and computed the final solution where damage is shown in each frame, with 5 FPS. If we increase the computing power with an efficient GPU with more RAM, we can increase the inference speed to 20-30 FPS which would make the solution a real-time one.

To decide whether the tool can be used for the next job, we compute the percentage of red pixels on the critical surface area. We average these percentage values over all frames and compute the overall damage percentage of the mandrel. We can then keep a threshold to identify which tools would require maintenance, which ones would be fit for the next job, and which would require a manual inspection when the damage percentage is neither too high nor too low.

For keeping a suitable threshold, we can analyze the defective mandrel videos and come up with a threshold that would be ideal for automatically deciding the reusage of the tool. So, by using the framework mentioned in the paper, we can make the inspection process, from identifying critical surface areas to damage detection to the final decision of tool reusability, completely automatic.

5. CONCLUSION

The framework provided in this paper can be used to digitally detect surface damage for any kind of tool whose geometry can be mathematically modeled. With the fixed setup, we only need the engineering designs of a tool to select the critical area and detect the damage on that area. Further, if we have gathered more data, we can also train the damage detection model for all kinds of available damages for a better decision-making process, as one kind of damage that occurs with less frequency can hamper the usability of the tool more than another kind of damage occurring in higher frequency. Also, with more data, we can increase the accuracy of the damage detection model and the solution gets better with each input. We are hopeful that this framework can be used to reduce the time in manual SME inspection and help them perform the task in real-time.

NOMENCLATURE

<i>SME</i>	Subject Matter Expert
<i>PASCAL</i>	Pattern Analysis, Statistical Modelling, and Computational Learning
<i>3D</i>	3 Dimensional
<i>2D</i>	2 Dimensional
<i>IoU</i>	Intersection over Union
<i>4D</i>	4 Dimensional
<i>RGB</i>	Red Green Blue
<i>CNN</i>	Convolutional Neural Network

GB Gigabyte
RAM Random-access memory
GPU Graphics Processing Unit
FPS Frames Per Second

REFERENCES

- Berveglieri, A., & Tommaselli, A. (2018). Reconstruction of Cylindrical Surfaces Using Digital Image Correlation. *Sensors* 18, no. 12: 4183. <https://doi.org/10.3390/s18124183>.
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- Canny, J.F. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8, 679-698.
- Collins, R. (2007). Camera Projection (Extrinsics). CSE/EE486 Computer Vision I. Fall 2007. Penn State University. Lecture.
- Collins, R. (2007). Camera Projection (Intrinsics). CSE/EE486 Computer Vision I. Fall 2007. Penn State University. Lecture.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database, *Proceeding of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (248-255), June 20-25, Miami, FL, USA. doi: 10.1109/CVPR.2009.5206848.
- Gruen, A. W. (1985). Adaptive Least Squares Correlation: A Powerful Image Matching Technique. *South African Journal of Photogrammetry, Remote Sensing and Cartography*, vol. 14, pp. 175–187.
- Gruen, A. (1996). Least square matching: A fundamental measurement algorithm. In *Close Range Photogrammetry and Machine Vision*. Bristol, UK: Whittle Publishing, pp. 217–255.
- He, K., Zhang, X., Ren, S., & Sun, J., (2016). Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, (770-778), June 27-30, Las Vegas, NV, USA. doi: 10.1109/CVPR.2016.90.
- Kingma, D. P., Ba, J., (2015). Adam: a method for stochastic optimization, *Proceedings of International Conference on Learning Representations*, May 7-9, San Diego, CA, USA.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *Proceedings of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, (234-241), October 5-9, Munich, Germany. doi: 10.1007/978-3-319-24574-4_28.
- Tan, M., & Le, Q. (2020) EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97:6105-6114.

BIOGRAPHIES



Gautam Kumar Vadisala is a Lead Data Scientist at Schlumberger Technology Center in Pune, India. He has his bachelor's degree (B.Tech.) in Electronics and Electrical Engineering from the Indian Institute of Technology Guwahati. His main interests are Computer Vision, Deep Learning, Machine Learning, and Data analytics.



Anurag Singh Rawat is a Machine Learning Engineer at Schlumberger Technology Center in Pune, India. He has his bachelor's degree (B.E) in Computer Science from Rajiv Gandhi Technical University. His area of focus is Machine Learning based scalable solutions on the cloud.



Abhishek Dubey leads the AI Product Development at Schlumberger Technology Center in Pune, India. He has an MS degree in Computer Science from the Indian Institute of Science, Bangalore. His area of expertise is building large-scale, end-to-end Machine Learning Products on the cloud. His main area of research is Deep Learning for structured & unstructured data and prognostics & health management.



Gareth Yen Ket Chin is a Senior Mechanical Engineer at Schlumberger. He has his bachelor's degree (B.E.) in Mechanical Engineering from the National University of Singapore. His main interests are Industry 4.0 Revolution, Data-driven Prognostic and Health Management (PHM) systems with Machine Learning (ML), and Deep Learning (DL) techniques.

Fabio Abreu is a Manufacturing Support Manager at Schlumberger. He has a master's degree in Reliability Engineering from the University of Tennessee – Knoxville.

APPENDIX

Although we haven't used it in our final solution, we have worked on a method that processes the video and creates an image that displays the total surface area of the mandrel. This method unwraps the whole mandrel surface and displays it on an image that would be ideally impossible to visualize at one time.

To achieve that, we process the video into frames and locate the mandrel in each frame using the previously mentioned

Camera Projection method. Then we extract a small horizontal strip of about 10 pixels (depending upon the length of the video) around the mandrel central surface in each frame and append them to create an image that displays the overall surface area. We use some reference points such as holes, grooves, etc. to mark a complete rotation of the mandrel in the video and not to over-represent the mandrel area on the unwrapped image. For example, if we assume the holes on the mandrel surface as reference points for rotation comparison, we note down the location of these holes in the first frame and stop the solution when we encounter the same holes again within some pixel distance of the original location.

For better damage detection, we can horizontally segment the unwrapped image into segments of 5 or 10 and combine the individual segment outputs for final damage detection of the unwrapped image. The final output can be seen in Figure A-18 where white vertical lines enclose the critical surface area. The red pixelated area within these lines shows the damage on the surface. One thing to note is that since we took a small strip around the mandrel center, our critical surface boundaries would not have a conical shape. Also, we can observe that some holes on the mandrel surface appear expanded, and others appear shrunk. This occurs because of the change in rotation speed within different points of the mandrel rotation. When the rotation speed decreases relatively, we will get more surface area in the unwrapped image and vice versa.

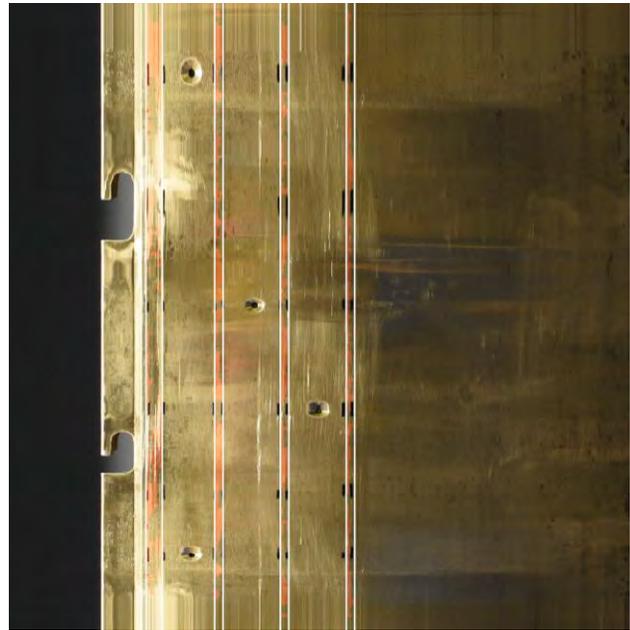


Figure A-18. Damage on the Unwrapped Mandrel

Here too, we can take a percentage of red pixels within the critical surface and create a threshold to come up with a decision on using the mandrel for the next job or not. If the mandrel doesn't maintain a fixed speed, we will not obtain an even spaced image of all its contents. Because of this reason, we did not select this method to come up with the final decision of reusing the mandrel as we would give more preference to the damage where the mandrel was recorded relatively slower than the other parts. But if the rotation is fixed, we can see the even spaced points across the unwrapped image. Nevertheless, this method can be useful to extract the surface area of the cylindrical objects in videos and further detect the damage on the surface.

State of Health Forecasting of Heterogeneous Lithium-ion Battery Types and Operation Enabled by Transfer Learning

Friedrich von Bülow¹, and Tobias Meisen²

¹*Volkswagen AG, Berliner Ring 2, 38436 Wolfsburg, Germany
friedrich.von.buelow@volkswagen.de*

²*Institute of Technologies and Management of the Digital Transformation, University of Wuppertal,
Rainer-Gruenter-Str. 21, 42119, Wuppertal, Germany
meisen@uni-wuppertal.de*

ABSTRACT

Due to the global transition to electromobility and the associated increased use of high-performance batteries, research is increasingly focused on estimating and forecasting the state of health (SOH) of lithium-ion batteries. Several data-intensive and well-performing methods for SOH forecasting have been introduced. However, these approaches are only reliable for new battery types, e.g., with a new cell chemistry, if a sufficient amount of training data is given, which is rarely the case. A promising approach is to transfer an established model of another battery type to the new battery type, using only a small amount of data of the new battery type. Such methods in machine learning are known as transfer learning. The usefulness and applicability of transfer learning and its underlying methods have been very successfully demonstrated in various fields, such as computer vision and natural language processing. Heterogeneity in battery systems, such as differences in rated capacity, cell cathode materials, as well as applied stress from use, necessitate transfer learning concepts for data-based battery SOH forecasting models. Hereby, the general electrochemical behavior of lithium-ion batteries, as a major common characteristic, supposedly provides an excellent starting point for a transfer learning approach for SOH forecasting models. In this paper, we present a transfer learning approach for SOH forecasting models using a multilayer perceptron (MLP). We apply and evaluate it on the method presented by von Bülow, Mentz, and Meisen (2021) using five battery datasets. In this regard, we investigate the optimal conditions and settings for the development of transfer learning with respect to suitable data from the target domain, as well as hyperparameters such as learning rate and frozen layers. We show that for the transfer of a SOH forecasting model to a new battery type it is more beneficial to have data of few old batteries, compared to data of many

new batteries, especially in the case of superlinear degradation with knee points. Contrarily to computer vision freezing no layers is preferable in 95% of the experimental scenarios.

1. INTRODUCTION

Due to the mobility transition to electric vehicles worldwide, the battery's state of health (SOH) gains interest of customers and consequently by research and industry. The SOH reflects the battery ageing which depends on the usage and environmental conditions of the battery. SOH forecasting enables e.g. fleet managers of battery electric vehicle (BEV) fleets to optimize their operational strategy w.r.t. battery ageing. In addition, by forecasting the replacement time of old BEVs due to battery degradation the transition to a new vehicle type can be supported. Nevertheless, establishing such services, requires reliability and availability in the underlying forecasting models. For example, when launching new BEVs with a new battery type, a usable model satisfying the aforementioned requirements is required.

In this regard, both von Bülow et al. (2021) and Richardson, Osborne, and Howey (2019) have published reliable models for SOH prediction, though both are data-intensive. However, this is not justifiable for application in changing environments in most cases, as new data to build a reliable model using these methods cannot be collected in a temporal manner. Hence, due to the aforementioned needed availability their wide acceptance is currently limited. This is especially problematic for new battery types whose data availability is limited in the initial phase: First, only few laboratory battery ageing test are conducted which suffer under limited comparability to real-world operational conditions (Nuhic, Terzimehic, Soczka-Guth, Buchholz, and Dietmayer 2013; Sulzer et al. 2021; von Bülow and Meisen 2022). Second, usually battery cells, but not packs, systems, or modules are tested in the laboratory. Third, potentially only a few prototypes or endurance tests may have been operated using the new cell type.

Friedrich von Bülow et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

One obvious solution to extend this limited data base, is further extensive data generation either in the laboratory or using endurance test vehicles. Data synthesis or augmentation from physical ageing models still needs validation from laboratory tests. Furthermore, we consider an adaption of feature values of ageing data and knowing the resulting ageing as too difficult. The complexity of this causal relationship is the main reason why data-driven models are considered for SOH forecasting. Nevertheless, these solutions are expensive and difficult as battery ageing is a lengthy process. In their work, von Bülow et al. (2021) proposed a solution by transferring an established model for battery ageing of another battery type to the new battery type, as soon as a small amount of data of the new battery type has been gathered. Such methods, summarized under the term transfer learning, have been successfully applied in different domains, like computer vision (Shao, Zhu, and Li 2015) and natural language processing (NLP) (Ruder, Peters, Swayamdipta, and Wolf 2019). The application of transfer learning for battery SOH forecasting models is a crucial part, as there are differences in batteries like the nominal capacity, the cell cathode materials as well as the applied load due to usage. However, the general electrochemical behavior of lithium-ion batteries is a major common characteristic which supposedly provides an excellent starting point for transfer learning. As von Bülow and Meisen (2022) state, until now transfer learning has not yet been applied to battery SOH forecasting.

In this regard, we contribute a transfer learning training process for SOH forecasting using a multilayer perceptron (MLP) with comprehensive experiments on five different known public datasets. We use the model presented by von Bülow et al. (2021) which showed very good results without limited availability of data. In the following, we examine when and how to transfer for. “When to transfer” concerns data availability in the target domain. I.e. the number of samples and their distribution by quantity and age of batteries. This shall enable practitioners to estimate the amount of data required for a successful transfer. “How to transfer” concerns the transfer method (i.e. freezing of layers of the pre-trained source model).

The remainder of this paper is structured as follows: Section 2 introduces the state of the art of battery components, their ageing and transfer learning including its applications in computer vision and battery ageing. In Section 3, the methods for SOH forecasting and transfer learning are explained. The used data basis is presented in Section 4. Subsequent, we present and discuss our results in Section 5. Section 6 concludes our work.

2. STATE OF THE ART

2.1. Lithium-Ion Battery Components and Ageing

The major components of a lithium-ion battery cell are: A negative electrode (anode), a positive electrode (cathode), the ion-conducting electrolyte, and the electrically insulating separator. For a schematic representation of a typical lithium-ion battery cell and information on the operating principle, interested readers are referred to (Keil 2017; Leuthner 2018; von Srbik 2015). The traditional cathode material has been lithium cobalt oxide (LCO). Alternatives are lithium nickel manganese cobalt oxide (NMC) and lithium iron phosphate (LFP) which have advantages over LCO regarding safety, cost, and size. Nevertheless regarding ageing, lithium-ion battery cells with different materials have different ageing characteristics (Vuorilehto 2018). Thereby, battery ageing is usually measured by the degradation of SOH. The SOH can be described by the internal resistance (SOH_R) and the remaining capacity (SOH_C) (Chen, Lü, Lin, Li, and Pan 2018; Waag, Fleischer, and Sauer 2014). The relative change of internal ohmic resistance compared to a new battery is the SOH_R . The capacity-based SOH_C is the remaining capacity $C(t)$ relative to the initial capacity of a new battery, also called nominal capacity C_{nom} (Lipu et al. 2018):

$$SOH(t) = SOH_C(t) = \frac{C(t)}{C_{nom}}. \quad (1)$$

In the following, we focus on the SOH_C , for simplicity referred to as SOH.

Battery ageing can be structured into two causes considered in this paper: Calendar ageing and cyclic ageing. Calendar ageing is associated with the storage of batteries, meaning no charging or discharging is applied. Hence, it is also called passive ageing. Cyclic aging corresponds to the impact of battery usage on the SOH, i.e. ageing due to charging and discharging (Gewald et al. 2020).

High temperatures (T) and high state of charges (SOC) are causing fast battery calendar and cyclic ageing (Matadi et al. 2017). For example, a high SOC over 80% accelerates solid electrolyte interphase (SEI) growth (Barré et al. 2013). Other stressors accelerating battery ageing are high charge and discharge C-rates¹ as well as a high ΔSOC (Gewald et al. 2020; Marongiu, Roscher, and Sauer 2015). Known battery stressors are qualitatively displayed in Table 1.

For the datasets used in this work, two types of battery ageing trajectories are relevant: Linear and superlinear degradation. Linear degradation is characterized by a constant aging rate over the whole battery life (e.g. NASA random in Appendix Figure 9). In contrast, batteries with superlinear degradation first age slowly, but change to accelerated ageing after the

¹ C-rate in [1/h]=[A/Ah] is the current relative to the nominal capacity C_{nom} .

Table 1. Aging mechanisms and their accelerating stressors (Birkel 2017; Nguyen 2019)

Battery component	Aging mechanisms	Accelerated by
Anode	Lithium plating	↑ C-rate, ↓ T, ↑ SOC
	Electrolyte decomposition	↑ T, ↑ SOC
	SEI formation	↑ & ↓ SOC
	SEI decomposition	↑ C-rate, ↓ T
	SEI growth	↑ T, ↑ SOC
	Structural disordering	↑ C-rate, ↑ & ↓ SOC
	Corrosion and loss of electrical contact	↓ SOC
Separator	Blocked pores (Separator and Electrodes)	↑ T, ↑ SOC
Cathode	Dissolving of transition metals	↑ T,
	Binder decomposition	↑ T, ↑ SOC
	Structural disordering	↑ C-rate, ↑ T
	Corrosion and loss of electrical contact	↑ C-rate, ↑ T, ↑ SOC

knee-point (e.g. Data-Driven in Appendix Figure 9) (Attia et al. 2022; Fermin-Cueto et al. 2020).

2.2. Transfer Learning

Transfer learning is a learning paradigm in machine learning that utilizes knowledge previously attained in one domain to solve a task in a novel domain (Pan and Yang 2010).

A domain $D = \{X, P(X)\}$ is defined by a feature space X and a marginal probability distribution $P(X)$, where $X = \{x_1, \dots, x_n\}, x_i \in X$. X represents a particular sample that is made up of different observations x_i which all lie in the feature space X .

A task $T = \{Y, f(\cdot)\}$ is defined by a label space Y and an objective predictive function $f(\cdot)$, which is learned from the training data. $f(\cdot)$ can be seen as $P = (y|x)$ from a probabilistic perspective.

In general, we assume that there is source domain data as $D_S = \{(x_{S_1}, y_{S_1}), \dots, (x_{S_n}, y_{S_n})\}$, where $x_{S_i} \in X_S$ is an input vector and $y_{S_i} \in Y_S$ is the corresponding output vector. The target domain data can be denoted as $D_T = \{(x_{T_1}, y_{T_1}), \dots, (x_{T_n}, y_{T_n})\}$, where $x_{T_i} \in X_T$ is an input vector and $y_{T_i} \in Y_T$ is the corresponding output vector. (Pan and Yang 2010)

Using these formal definitions, transfer learning can be defined in the follow way:

“Given a source domain D_S and learning task T_S , target domain D_T and learning task T_T , transfer learning aims to help improve the learning of the target predictive function $f(\cdot)$ in D_T using the knowledge in D_S and T_S , where $D_S \neq D_T$, or $T_S \neq T_T$.” (Pan and Yang 2010)

When applying transfer learning, Torrey and Shavlik (2010) identify three measures by which learning might be improved due to a transfer. These are visualized in Figure 1

at the exemplary model performance of a regression task measured by the root mean squared error (RMSE): Start, convergence speed and the asymptote of the model’s performance may improve. First, the initial RMSE before any training with data from the target domain might be lower with transfer learning than without. This occurs when a model, i.e. an artificial neural network (ANN) trained on source domain data provides a better starting for learning than a randomly initialized model. Second, transfer learning can increase the convergence speed, i.e. it can decrease the training time to fully learn the target task. Third, the final performance achievable in the target task with transfer learning can be lower than without transfer learning. The lower start and lower asymptote can easily be measured by the model performance at the start and end of training respectively. However, measuring the higher convergence speed is more difficult. A popular metric is the area under the learning curve (AULC). For the convergence speed we are interested in the improvement of the model performance. Thus, the exponent

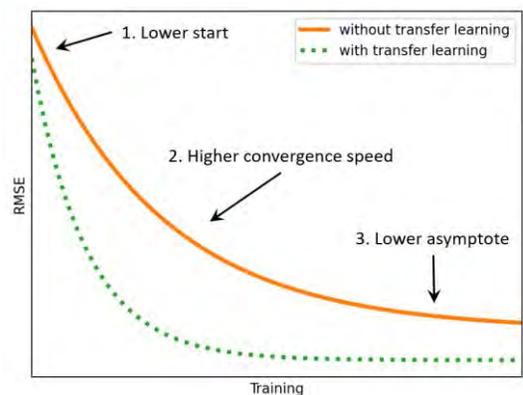


Figure 1: Measures in which transfer might improve learning: 1. lower start, 2. higher convergence speed and 3. lower asymptote (adapted from Torrey and Shavlik 2010).

of a power-law fit is more suitable (Bertoldi, Cettolo, Federico, and Buck 2012; Vierung and Loog 2021). Mathematically, this measure is concerning relative convexity (Palmer 2003). However, overall we consider the lower asymptote as the most important measure because the final model performance after transfer learning is the most relevant. Thus, we focus our experiments and their discussion on the final model performance.

Further, Pan and Yang (2010) name three main concerns relevant when implementing transfer learning: What, how and when to transfer. What to transfer concerns the part of knowledge that is transferable across domains or tasks. On the one hand, the knowledge may be domain or task specific. On the other hand, it may be common across different domains and tasks. In the latter situation transfer learning may be beneficial. How to transfer concerns the learning algorithm for transfer learning, such as fine tuning, layer-wise freezing of an ANN or domain adaptation. Lastly, when to transfer also includes the question of when not to transfer to avoid a negative impact on the performance in the target domain which is known as negative transfer

2.2.1. Computer Vision

Transfer learning is well established in the field of computer vision. For example, for image classification pre-trained convolutional neural network (CNN) models like VGG-16, ResNet50, and Inceptionv3 are popular. These models were trained on large datasets, like the VGG-16 on 1.3 million images with 1,000 classes (Simonyan and Zisserman 2014).

Yosinski, Clune, Bengio, and Lipson (2014) stated that on the first convolutional layers CNNs learn abstract features such as edges and corners that are relevant for many datasets and tasks. When only a much smaller target dataset is available, learning such abstract features with a good generalization might be difficult. In this case, the first layers of the pre-trained CNN are useful as feature extractor and improve generalization. Building on these features, some of the higher convolutional layers and the classifier part of the CNN generate specialized features.

In addition, features from CNNs are suitable for many computer vision tasks (Razavian, Azizpour, Sullivan, and Carlsson 2014). The major advantage of transfer learning in computer vision is the elimination of the lengthy training process. Rawat and Wang (2017) provided a comprehensive overview of the application of CNNs to visual tasks.

2.2.2. Battery Ageing

Compared to the application of transfer learning in computer vision, pre-trained models related to battery ageing or SOH forecasting do not exist. This is potentially due to the fact that there are only few publicly available battery ageing datasets. In the field of battery ageing standardized tasks like RUL prediction and SOH estimation exist, just like image

classification and image segmentation in computer vision. However, input features are very different regarding complexity and aggregation which complicates providing pre-trained models.

For transfer learning applied to SOH forecasting no direct related work exists. However, there exist two works in the context of transfer learning and battery ageing: Azkue, Lucu, Martinez-Laserna, and Aizpuru (2021) trained a baseline MLP using only calendar ageing data of 30 NMC cells. Then they apply transfer learning with a reduced amount of another LFP dataset. As benchmark they trained a model solely on the LFP dataset. They achieved a better performance with two LFP cells and transfer learning than without transfer learning and five LFP cells. Moreover, their results indicate improved generalization with transfer learning. However, they only compared to one benchmark model and use only one target dataset. In addition, the transfer is only examined for calendar ageing, not cyclic ageing.

Shen, Sadoughi, Li, Wang, and Hu (2020) trained a CNN for capacity estimation of lithium-ion batteries on a source dataset of ten years of cycling ageing data. All five convolutional layers and the three fully-connected layers of the pre-trained model were transferred. The best performance was achieved, when fine-tuning all layers providing the most capabilities for model adaption to the target domain. Compared to Azkue et al. (2021), Shen et al. (2020) vary the amount of target data available for fine-tuning. They find that training a benchmark model from scratch needs three times more training samples compared to transfer learning. Thus, transfer learning saves time and costs for data collection in their application.

In summary, these two papers indicate that transfer learning is a promising approach for battery ageing models. However none of these papers, considered the temporal sequence when data becomes available for training. This consideration is important to answer the question “when to transfer.” Further, each paper only used one target dataset. Thus, they cannot be used to compare the influence of different cell chemistry and battery operation on the success of transfer learning.

3. METHOD

In Section 3.1, a short overview of a method for SOH forecasting is given whose suitability for the task was shown (von Bülow et al. 2021). Building on this method for SOH forecasting, we introduce a training process for transfer learning in Section 3.2 to overcome the lack of training data for new battery types. This training process can also be applied to other machine learning methods in the field of battery ageing.

3.1. State of Health Forecasting

As mentioned in Section 2.1, battery ageing is perceived as a state change from a current $SOH(t_1)$ to a future $SOH(t_2)$ due

to ageing causes. The ageing causes are encoded in the battery operational data which consists of multidimensional time series signals of c-rate, temperature, and SOC. As depicted in Figure 2 first, this battery operational data is used to extract stressor table data of battery stressor types which are known to induce battery ageing. Second, the flattened stressor table data is input of a machine learning (ML) model, that outputs the state change ΔSOH from a current $SOH(t_1)$ to a future $SOH(t_2)$. The SOH values are assumed to be known for the training data. The two parts of the proposed SOH forecasting method are explained in detailed by von Bülow et al. (2021).

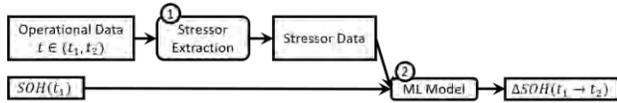


Figure 2: Model structure - stressor extraction (1) and ML model (2) (von Bülow et al. 2021)

3.2. Transfer Learning

First, a SOH forecasting MLP model in D_S , the source model, is (pre)-trained according to Section 3.1. Second, the source model is parametrically transferred to another domain D_T that has only little data available at transfer time.

We assume D_S to be a battery type with plenty of training samples available. Further, D_T represents a new battery type that has little available training samples at the point in time t of the transfer ($|D_T(t)| \ll |D_S|$). We assume that in time the amount of training samples of the new batteries increases ($|D_T(t_1)| < |D_T(t_2)|$ for $t_1 < t_2$). We preprocess the battery operational time series data according to von Bülow et al. (2021). For all datasets the same signal interval width and the same boundaries of the stressor tables in D_S and D_T are used so that $X_S = X_T$. However, for different operation and different battery types $P(X_S) \neq P(X_T)$ hold true because the distribution within the stressor tables and also $SOH(t_1)$ is different. As $P(X_S) \neq P(X_T)$, also the domains are different ($D_S \neq D_T$).

Regarding the learning tasks, we assume $Y_S = Y_T$ because we only output ΔSOH . However, we state that $f_S(\cdot) \neq f_T(\cdot)$ because the range of ΔSOH is different for the battery types in \mathcal{T}_S and \mathcal{T}_T given $X_S = X_T$. E.g. the target domain may be unbalanced with more new batteries. As $f_S(\cdot) \neq f_T(\cdot)$, also the tasks are different ($\mathcal{T}_S \neq \mathcal{T}_T$). As source and target task are different, we face the setting of inductive transfer learning. Thus, we need to use labeled data from the target domain to induce $f_T(\cdot)$ (Pan and Yang 2010).

3.2.1. What to Transfer: Common Characteristics

In the context of this work, “what to transfer” concerns the common characteristics of the relationship of current $SOH(t_1)$, battery operation (c-rate, temperature, and SOC) and the future $SOH(t_2)$.

3.2.2. How to Transfer: Layer Freezing

We accomplish knowledge transfer by a parametric transfer of the weights and biases from a source model to the target model. As common procedure in computer vision shown in Section 2.2.1, weights and biases of the model’s layers can be frozen, i.e. they do not change while further training. Thus, the knowledge learnt from D_S is preserved in the frozen layers. In the unfrozen layers, the weights pre-trained in D_S serve as starting point compared to randomly initialized weights. After freezing, we continue training the target model with the target training data. The parametric transfer has the transfer hyperparameter learning rate α , epochs, and number of frozen layers n_{frozen} . The number of frozen layers is counted from the input layer towards the output layer as the example in Figure 3 shows. For answering “how to transfer”, we examine how to optimize n_{frozen} for achieving the best result on the target test data.

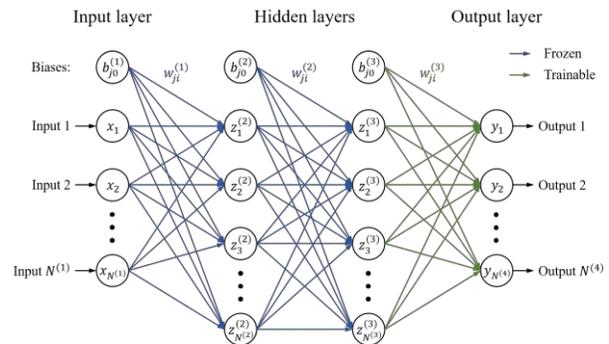


Figure 3: MLP with exemplary freezing of two layers (first and second). The output layer is trainable.

3.2.3. When to Transfer: Data Availability

“When to transfer” for battery ageing models refers to the temporal availability of target data. Thus, we artificially split the target data into training and validation data which is available at transfer time and test data which has not been recorded at transfer time. The test data will be input to the model when it runs in production, i.e. in prediction mode applied e.g. by automotive manufacturer or BEV fleet managers. Thus, model evaluation on the test data is a suitable measure for the success of the transfer. The amount of training target data is measured by the number of samples available for the transfer. The distribution is defined by the quantity and age of batteries, i.e. the maximum cycle number.

The target data is split according to three scenarios inspired by an automotive manufacturer that will introduce a BEV with a new battery type. The new battery type is the target domain D_T from a machine learning point of view. For each data split, first the order of the battery cells is shuffled. Then samples are added to the training and validation dataset according to the data split until the specified number of samples is reached. If multiple window lengths are present in the dataset, we sort ascendingly. The remaining samples

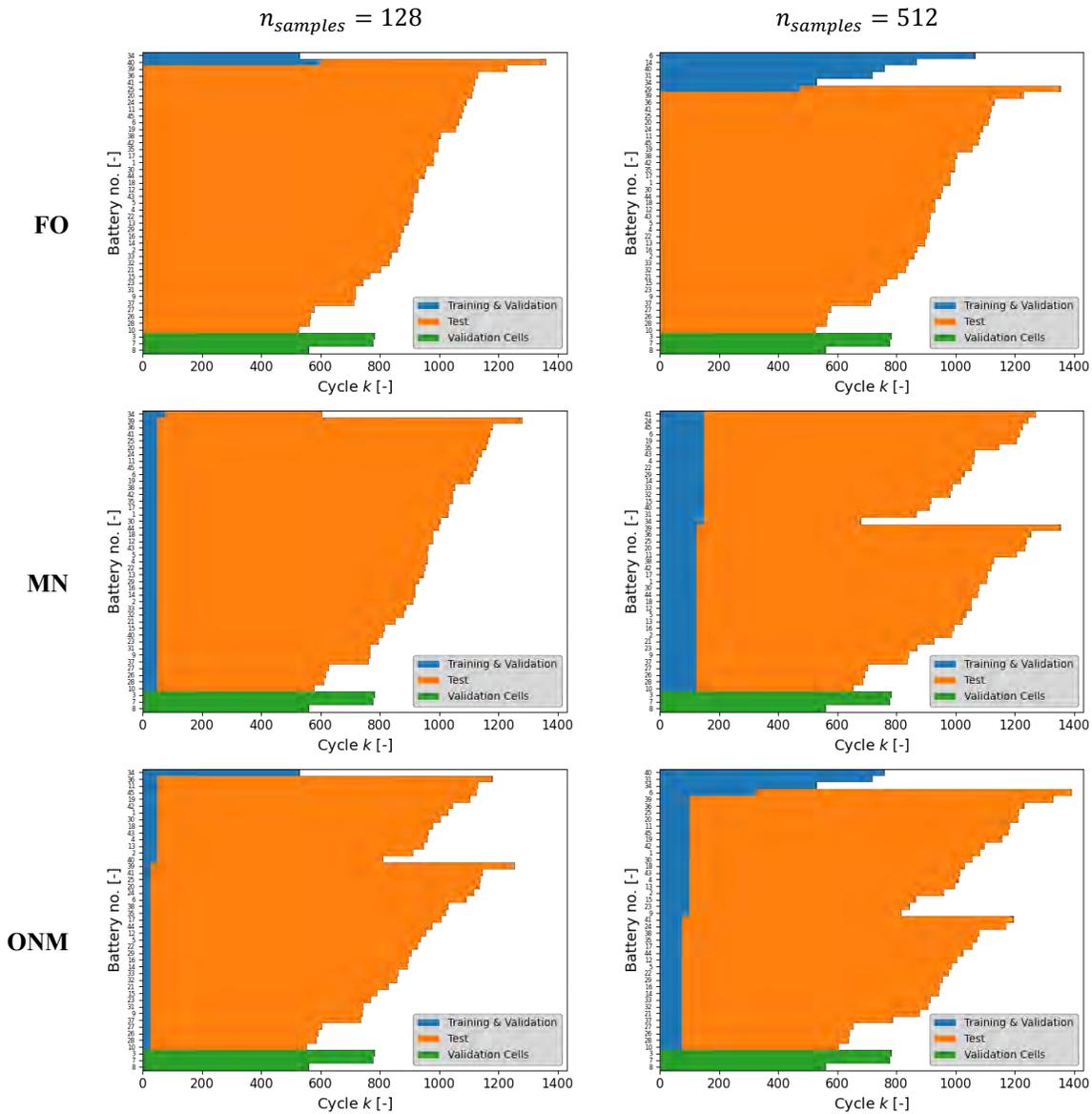


Figure 4: Exemplary data splits “few old” (FO), “many new” (MN), and “old new mixed” (ONM) with $n_{\text{samples}} = \{128; 512\}$ using the Closed-loop dataset and random seed 0.

compose the test dataset. The three data splits are shown exemplary in Figure 4:

First, the data split “few old” (FO) assumes that before the market launch of the new BEV type endurance tests are run: A few vehicles are intensively driven so that components like the batteries age quickly (Choi, Jung, Ham, and Bae 2011; Gassner 1984). Further, limited data from laboratory experiments is possibly available. Thus, target data of few, but old batteries is available for transfer learning.

Second, the data split “many new” (MN) assumes that after the market launch of a new BEV type after some time already many vehicles are driven by customers. However, these batteries have only aged very little or non-measurably. Thus,

target data of many, but new batteries is available for transfer learning.

Third, the data split “old new mixed” (ONM) assumes the availability of data from the first two data splits. Thus, target data of many, but new batteries and few, but old batteries is available for transfer learning.

3.2.4. Benchmarks

For comparing the performance of transfer learning with non-transfer learning approaches whilst having the same availability, we define three benchmarks with a different composition of training and validation data as shown in Table 2. All benchmarks are tested on the same test data like the

Table 2: Overview of benchmarks for transfer learning

Benchmark	Training and validation data	Test data
Source Only (SO)	D_S	
Source Target Mixed (STM)	D_S & D_T	D_T
Target Only (TO)	D_T	

MLPs with transfer learning according to the data split type and the number of samples available. Only the training and validation data differ. Source Only (SO) is the source model without further modifications tested on the target test data. Source Target Mixed (STM) uses data from source and target domain for training and validation. The Target Only (TO) model is exclusively trained and validated on target domain data. For comparability, all benchmarks are MLPs with the same network architecture as the source model, but the weights are randomly initialized. Only their learning rate and batch size are optimized.

4. DATA

We selected public datasets that first provide data of sufficient battery cells, second have overall the same as well as different battery types, and third have a variety of battery operation to examine transfer learning in different circumstances. We differentiate battery types among others regarding materials and nominal capacity. Regarding battery operation charging, discharging, and storage of the batteries are relevant. This variety enables us to examine the effect of different battery types and operation. An overview of the used datasets regarding battery operation and materials is given in Table 3.

In the following, for each dataset we first present a brief description. Then, if necessary, we describe applied data cleansing methods, if physical inconsistencies in the data like time jumps or measurement errors were found. Lastly, cells that were not used and the reasons for this decision are specified.

The following two procedures were applied to more than one dataset:

- For the NASA Random and Oxford dataset, the start of a new cycle is not given. Thus, we define cycle starts when the current in a new step switches from zero or below zero (storage or discharging) to higher than zero (charging) so that every cycle starts with charging. But we require the previous cycle to be at least a full equivalent cycle (FEC) before a new cycles starts.
- For ISEA, NASA Random, and Oxford, capacity measurements are conducted in regular intervals. The

SOH values are interpolated over time in between these capacity measurements.

4.1. Stanford Datasets

Severson et al. (2019) and Attia et al. (2020) present public datasets using the same battery cells with varying fast charging protocols. We refer to these two datasets by the paper’s name first using the dataset as "Data-Driven" and "Closed-Loop" respectively. These commercial LFP/graphite cells, manufactured by A123 systems (APR18650M1A), were cycled in a forced convection temperature chamber set to 30°C under varied fast charging conditions but identical discharging conditions of 4C. The cells have a nominal capacity of 1.1 Ah. The sampling rate is 1s.

The SOH values are defined by the discharged electric charge of the corresponding cycle. The SOC signal is calculated offline relatively to the discharged electric charge of the previous cycle.² SOH values corresponding to single cycles are identified as outliers when they deviate more than three local standard deviations from the local mean within a 30-element window of the neighboring SOH values. These values are interpolated linearly using the neighboring SOH values.

4.1.1. Data-Driven

The Data-Driven (DD) dataset (Severson et al. 2019) used in this work consists of 42 lithium-ion batteries belonging to the batch of 2017-05-12 cycled up to 80% SOH. All cells are charged with a two-step fast-charging protocol. This protocol has the format “C1(Q1)-C2”, in which C1 and C2 are the C-rates of the first and second CC steps (CC1 and CC2 respectively) and Q1 is the SOC at which the current switches. C1 and C2 range from 3 to 8C, while Q1 ranges from 15 to 80 % SOC.³ The second current step ends at 80% SOC, after which the cells charge with another CC step at 1C (CC3) followed by a CV phase.

The SOH signal is smoothed using a moving mean with a span of 15 data points because the SOH signal is noisy. The cells 1 to 5 of batch 2017-05-12 continued as cells 8 to 17 in batch 2017-06-30. Thus, we concatenated them in batch 2017-05-12. The thermocouples of cells 15 and 16 in batch 2017-05-12 were switched. This means that the temperature signals of these cells are not synchronized with the other cells’ signals. It was not possible for us to synchronize them. Thus, we do not use these cells. Neither do we use cells no. 1 and 19 as some of their cycles are up to 11.8 times longer as the previous cycles because the switching from charging to discharging happens late. This would skew the min-max-normalization. We use cells no. 3, 7, and 8 as validation cells which means that they are not part of the training, validation or test set.

² The discharged electric charge is specified by the end value of the signal "Qd".

³ C1 and C2 ∈ {3, 3.6, 4, 4.4, 4.8, 5.4, 6, 7, 8}C. Q1 ∈ {15, 25, 30, 35, 40, 50, 60, 70, 80} % SOC.

4.1.2. Closed-Loop

The Closed-Loop (CL) dataset (Attia et al. 2020) used in this work consists of 45 lithium-ion batteries belonging to the batch of 2019-01-24 cycled up to 22.8 % SOH. All cells are charged with a three-step fast-charging protocol. This protocol has the format "CC1-CC2-CC3", in which CC1, CC2, and CC3 are the C-rates of the first, second, and third CC steps that end at 20%, 40%, and 60% respectively. CC1, CC2 and CC3 each range from 3.6 to 8C. A fourth parameter, CC4, is dependent on CC1, CC2, CC3, and the charging time. CC4 ranges from 2.68 to 4.755C.⁴

The SOH signal is smoothed using a moving mean with a span of 25 data points because the SOH signal is noisy. We use cells no. 3, 7, and 8 as validation cells which means that they are not part of the training, validation or test set.

4.2. ISEA

The RWTH ISEA (ISEA) (Sauer 2021) battery dataset consists of 48 commercial lithium-ion battery cycled with the same profile under equal conditions to explore intrinsic cell manufacturing variability and small temperature differences in battery packs during operation. These cylindrical

Table 3: Overview of the used datasets and their relevant characteristics

Dataset		Stanford Data-driven (DD)	Stanford Closed-Loop (CL)	ISEA	NASA randomized (random)	Oxford
Source		(Severson et al. 2019)	(Attia et al. 2020)	(Sauer 2021)	(Bole, Kulkarni, and Daigle 2012)	(Raj, Wang, Monroe, and Howey 2020)
Battery ageing	Number of cells	42	45	48	26	27
	Life time range [cycles]	532-2,235	530-1,231	1,410-1,872	199-1,188	300-1,681
	Min. SOH [%]	80-88	22-76	50	50-86	2-86
Battery type	Nominal capacity [Ah]	1.1	1.1	$\cong 1.85$	2.1	3
	Nominal voltage [V]	3.3	3.3	3.7	?	3.6
	Voltage limits [V]	2.0-3.6	2.0-3.6	3.0-4.1	3.2-4.2	2.5-4.2
	Packaging style	18650 (cylindrical)	18650 (cylindrical)	18650 (cylindrical)	18650 (cylindrical)	18650 (cylindrical)
	Cathode material	LFP	LFP	NMC	LCO	NCA
	Anode material	Graphite	Graphite	Carbon	Graphite	Graphite
Battery operation	Max. charging c-rate	3-8C	3.6C-8C	2.2C	2.2C	0.5C
	Max. discharging c-rate	-4C	-4C	-2.2C	-2.4C	-0.5C
	Δ SOC	100%	100%	60% (SOC 20-80%)	Up to 100%	100 %
	Ambient temperature [°C]	30	30	25	20 & 40	24
	Storage time	0h	0h	0h	0h	5 or 10 d

⁴ CC1, CC2, and CC3 \in {3.6, 4.4, 4.8, 5.2, 5.6, 6, 7, 8}C. CC4 \in {2.68, 3, 3.652, 3.834, 3.94, 4.16, 4.252, 4.755}C.

NMC/graphite cells, manufactured by Panasonic/ Sanyo (UR18650E), were cycled at constant ambient temperature of 25°C. One cycle consists of 30 min discharging to 3.5 V and 30 min charging to 3.9 V, both currents limited to a maximum of 4 A. The charge turnover is about 1 Ah corresponding to cycles between approximately 20% and 80% SOC. Because of the voltage limits the charge turnover varies with the SOH of the cell, but the DOD in relation to the aged capacity is being kept nearly constant. The cells were graded into group C from the manufacturer and are drawn from the same production lot. Due to this factory selection the cells only have a mean capacity of approximately 1.85 Ah which we chose as nominal capacity. Capacity measurements are on average conducted every 165 cycles. The sampling rate is on average 0.1s.

The SOC signal calculation is orientated at the given boundaries of approximately 20% and 80% SOC for each cycle. Jumps of the time signal were corrected for cell 3 and 8. If the temperature signal is 0°C for a complete cycle, the ambient temperature of 25°C was set because this is the lowest physically possible temperature. This ensures that these states are considered in the stressor tables at least in the ambient temperature bin. If the temperature dropped to 0°C only for some time stamps, it was interpolated using the neighboring values because sudden temperature breaks are seen as impossible as the temperature only changes slowly. Cell 47 has a noisy temperature signal which was smoothed using an Fourier transform (FT) and moving mean. For some cells the last cycle length is 1,820 min instead of 60 min. For cell 29 and 18 the last 3 and 86 cycles respectively are very short. Because of these irregularities, we only use data corresponding to SOH higher than 50%.

4.3. NASA Randomized

NASA Ames Prognostics Center of Excellence Randomized Battery Usage Data Set (NASA Random) (Bole, Kulkarni, and Daigle 2012) consists of 26 battery cells of type LCO/graphite LG Chem. 18650 that were cycled in seven groups with different cycling protocols for each group (Details Appendix Table 7). The cycling protocols specify randomized sequences of current loads ranging from 0.5 A to 4 A. The sequences are randomized in order to better represent practical battery usage. The temperature was environmentally controlled. The sampling rate is 1s. After every fifty randomized discharging cycles, the capacity was determined (reference discharge cycles). These capacity measurements are executed twice.

Some capacity measurements were incomplete, resulting in a lower capacity value. Thus, if the capacity values deviate more than 0.1 Ah from each other, the higher values is chosen. The SOC signal calculation is orientated at the voltage limits after charging and discharging. We identified voltage jumps from 3.2 to 4.2 V combined with a time jump which are likely due to missing charging data. For best

correction we added values with the sampling rate of 1s of the dataset: The current so that battery is fully loaded with CC. Temperature and voltage are linearly interpolated. Else the average of both is used. Batteries RW2 and RW18 were not used because the temperature signal has many values of -4,000°C which lies below absolute zero.

4.4. Oxford

Oxford Path Dependent Battery Degradation Dataset (Oxford) (Raj, Wang, Monroe, and Howey 2020) consists of 27 battery cells of type Panasonic NCA/graphite 18650 that were cycled in four groups with different cycling protocols for each group. All cells were cycled with CC between 0% and 100% SOC. After a period of cycling, calendar ageing was performed at 90% SOC. The time ratio of cyclic to calendar ageing was 1:5 with different c-rates and storage SOC (Details Appendix Table 6). Compared to the other datasets, only the Oxford dataset contains calendar aging. The temperature was environmentally controlled at 24°C. Capacity measurements are conducted every 48 cycles. The sampling rate is 1s.

The SOC signal calculation is orientated at the voltage limits after charging and discharging. Again, we justify the replacement of temperatures below 0°C and Not a Number values (NaN) by the ambient temperature of 24°C, as we aim to minimize adulteration of the training data ($T_{\min} = 24^{\circ}\text{C}$, $T_{\max} = 36.6^{\circ}\text{C}$, $\Delta T = 25.2^{\circ}\text{C}$). Inconsistency like negative time deltas and time jumps of “TestTime” higher than the sampling rate with following constant time values were corrected by replacing the time signal with the sampling rate of 1s onwards. This sampling rate was determined from the consistent time signals. This adaption seems legitimate to us, especially for the time jumps, because the duration of the time jumps equals the product of sampling rate and the amount of following constant time values. The file no. 27 of battery number 9 from part 2, group 1 has some single current values around -10^{282} A during CC phases. These were replaced by the average of the two neighbor current values. Battery no. 1 from part 2, group 6 was not used because it is only exposed to calendar ageing which prohibits the identification of cycles. Battery no. 10 from part 1, group 2 was not used because it only contains a single cycle. The file no. 14 and 15 of battery no. 14 from part 1, group 3 are renamed and used for battery no. 15 of the same part and group because these are the only files of battery no. 14. At the same time exactly these file numbers are missing for battery no. 15.

5. RESULTS AND DISCUSSION

5.1. Design of Experiments

We aim at answering when and how to transfer a SOH forecasting model from the source to the target domain. For developing the parametric transfer the essential model

hyperparameters are the learning rate α and the number of frozen layers n_{frozen} . In addition, we compare the transfer learning to the benchmarks proposed in Section 3.2.4. Further, we examine the number of samples and their distribution over different batteries and ageing states (data split type). Model users are also interested in how the number of samples and data split type are influenced by the similarity of source and target dataset.

The target datasets for training and validation have 128, 256, 512, 768, and 1024 samples. This is limited by the NASA Random dataset which has only 1183 samples. We use a learning rate in the common range of 0.001, 0.0001, and 0.0001. Training did not converge with higher learning rates. We use the RMSE as evaluation metric because it is common for regression problems and, compared to the MSE, has the same unit as the predicted output value.

The source dataset of all experiments is the DD dataset which was also used in previous works (von Bülow et al. 2021). The source dataset contains 3127 training samples and each 391 validation and test samples. The source model is obtained following the method presented in previous work using the DD dataset again (von Bülow et al. 2021). The DD dataset provides an ideal starting point for our experiments because of its similarity to CL. We use fine 2D stressor table, variant A because this showed the best result in previous work (Appendix Table 4 and Table 5). Further, we assume that the fine signal interval width enables the model to learn the target data better. Batteries in the Oxford and NASA random dataset have a lifetime below 400 cycles. This prohibits window lengths of 400 and 530 (W5 and W6 in previous work) as well as the corresponding grouped datasets W10 to W12. Further, a grouped window length of {25; 50; 500} cycles (named W9 in previous work) showed promising results regarding generalization in the source domain. Thus, we chose W9 and a window shift of $w_s = 25$. The hyperparameters, model complexity and model performance of the source model on the source data are shown in the Appendix in Figure 9.

For better comparability, the benchmark MLPs use the same amount of layers and neurons as the source model. For TO and STM, only the learning rate $\alpha \in \{10^{-N} | 3 \leq N \leq 5\}$ and the batch size $bs \in \{16; 32; 64; 128\}$ are hyperparameters of a grid search. For SO, the source model is simply run in prediction mode as it is already trained only with the source data.

Compared to previous work (von Bülow et al. 2021), the min-max normalization is adapted to be suitable for the transfer learning. In previous work, min-max normalization was executed feature-wise using the source training data, i.e. each feature separately was normalized based on its min and max value in the source training data. The minima in the source and target data are mostly zero because the dwell time for every stressor type is zero. However, the max values of the target training data are different as stated in Section 3.2

($P(X_S) \neq P(X_T)$). Furthermore, determining the maxima of the features of the test target data is not possible at transfer time, because the test target data becomes only available in the future. Thus in this work, min-max normalization is executed in two groups: First, for the $SOH(t_1)$ and second for all other features, i.e. the stressor values from the stressor tables. We use the maximum of all stressor values of the source training data for normalization. Further, we keep the same scaling from the source data for all target data to ensure that the weight of the stressor values is the same for all features. In addition, this eases the model's transfer because the model does not need to adapt to a different normalization of the target data.

Experiment 1: How to transfer? Freezing & Learning rate

In general, we expect n_{frozen} and α to jointly influence the convergence speed and also the final target model performance. n_{frozen} is a measure how much information from the source domain is kept at transfer time. Thus, we have a special interest in it as it also is a measure of similarity between source and target domain. We assume that a frozen model will perform well, if the domains are similar. Contrarily, if less information on the future ageing is given in the target domain training data, freezing is also beneficial to prevent negative transfers.

Experiment 2: When to transfer? Data splits & no. of samples

After examining the selection of the best model regarding n_{frozen} and α in Experiment 1, the question “when to transfer” concerns the suitability of the target data specified by the three data splits FO, MN, and ONM as well as the number of samples.

Experiment 2a: Closed-Loop

As visible in Table 3, the batteries in DD and CL have the same battery characteristics. Only the battery operation deviates regarding the charging protocols. Thus, DD and CL are similar domains, providing an ideal situation for transfer learning.

Experiment 2b: ISEA, NASA random, and Oxford

The transfer to the ISEA dataset provides a change in cathode material from LFP to NMC and a smaller SOC range. Batteries in automotive applications like BEVs experience variable discharging and not simple CC discharging like batteries in laboratory operation (von Bülow and Meisen 2022). Compared to the other datasets, in the NASA random dataset not CC discharging, but stepwise randomly changing discharge rates are applied. Not only cyclic ageing, but also calendar ageing is important in automotive applications for battery ageing which is considered by the Oxford dataset.

Because of these differences that influence the stressor data ($\subset P(X_T)$) and the predictive function $f_T(\cdot)$, we expect these

three datasets to be more challenging for successful transfer learning than the CL dataset. Furthermore, the NASA random and Oxford datasets are a step towards applying the SOH forecasting model and its transfer to BEV operational ageing data.

Each non-test target dataset was randomly split for training and validation by the ratio of 90:10. The validation MSE was set as metric for early stopping with patience of 5 epochs and minimum delta of 0 to avoid overfitting on the training target data. We apply a maximum of 90 epochs. The constant hyperparameters from source model training are not changed: Optimizer Adam, MSE as loss function, and a linear activation function of the output layer. We also opted for keeping the kernel regularizer for the transfer to prevent overfitting. Version 2.8.0 of TensorFlow was used as backend including version 2.8.0 of Keras.

5.2. Evaluation of Experiments

When selecting a model regarding the hyperparameters n_{frozen} and α for a given target dataset, the validation and the test target RMSE are possible selection criteria. Both have a different distribution of the output values because of the temporal data split specified in Section 3.2.3. which leads to different ageing rates for samples from the beginning of life (BOL) compared to the end of life (EOL) (see Appendix Figure 9). Due to these assumptions of data availability for the model selection at transfer time only the validation target set is available, even though we are overall interested in the test target RMSE which indicates the model’s performance when run in production.

Result experiment 1: How to transfer? Freezing & Learning rate

In experiment 1 we do not only evaluate which n_{frozen} and α are optimal in the majority of scenarios, but we also evaluate the reduction of model performance on the test dataset due to the selection criteria only being the validation RMSE. Exemplarily, Figure 5 shows the model’s RMSE of all three α with different n_{frozen} for ISEA, MN with 128 samples. Selecting by the test RMSE (rectangles) $\alpha = 0.001$ and $n_{frozen} = 7$ would be optimal, but by the validation RMSE (crosses) $\alpha = 0.0001$ and $n_{frozen} = 0$ is best.

As depicted in Figure 6 for all four target datasets and selected by the validation RMSE, freezing no layer is the best choice in 95 % of the data splits ($n_{frozen} = 0$). However, when selecting by the test RMSE this would only be the case in 45 % of the data splits. 18 % of all data splits have a better test RMSE with freezing the first layer instead of none. This indicates that in the majority of the data splits no general features have been learnt from the source domain. Compared to the state of the art procedure in the field of computer vision this is potentially due to the limited amount of source training data. $n_{frozen} = 10$ selected by the test RMSE is only optimal

to prevent a negative transfer in the case of MN and CL as further discussed in Experiment 2.

Further, for α the selection based on validation and target RMSE diverges even more as shown in Figure 7. For 72 %

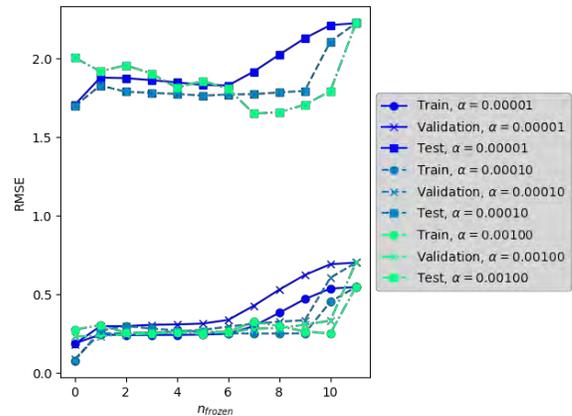


Figure 5: Exemplary model selection (ISEA, MN, 128 samples)

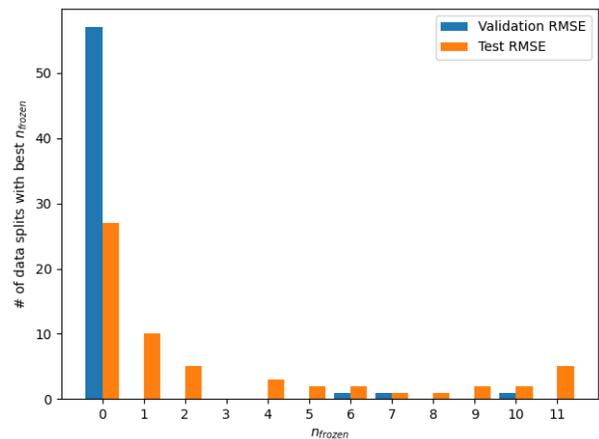


Figure 6: Best n_{frozen} selected by validation and test RMSE of all four target datasets

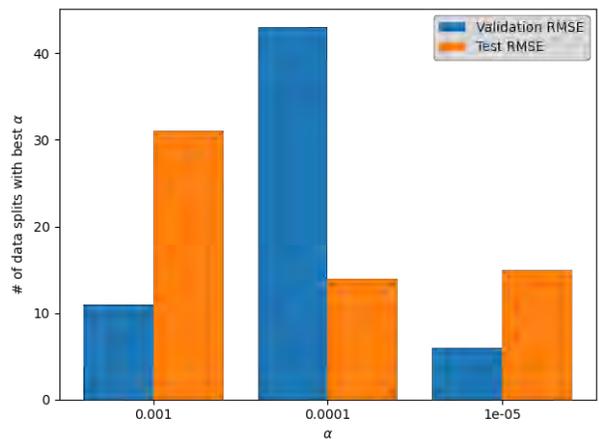


Figure 7: Best learning rate α selected by validation and test RMSE of all four target datasets

of the data splits 0.0001, which is also the source model's α , is the best α when selected by the validation RMSE, but for 60% of the data splits 0.001 is the best when using the test RMSE. Overall, these discrepancies lead to an average deviation of the test RMSE when selecting n_{frozen} and α by the validation RMSE of 23% with a standard deviation of 28%. Unfortunately, we have to accept this error because we cannot access the test RMSE itself for model selection. In the following, for each data split we select the best model based on the validation RMSE.

Experiment 2: When to transfer? Data splits & sample no.

Figure 8 depicts the performance of the best model selected by validation RMSE on training, validation, and test data as well as the SO, TO, and STM benchmarks of all four target datasets. We first analyze CL as target dataset and then the remaining three target datasets.

Experiment 2a: Closed-Loop

Figure 8a) shows for the CL target dataset FO that the training and validation RMSE are on the same level as the source RMSE (see Appendix Table 8 for comparison). The test

RMSE is higher overall and just acceptable for an application of the model e.g. by fleet managers who are interested in forecasting degradation from 100% down to approximately 80%. The test RMSE is decreasing once more data becomes available for training. This is coherent with our expectations. Independently of the amount of data only STM can compete with transfer learning. TO only reaches a similar model performance with 1,024 training samples.

For MN, we observe a negative transfer that leads to a RMSE of around 5 which is too high for a practical model application. This is caused by a lack of degradation information about the knee point. Even with 1,024 samples not sufficient information about ageing at EOL and the knee point is provided because 1,024 samples only correspond to the first 250 cycles and a minimum SOH of 93%, but the earliest knee point of CL is at around 500 cycles (Appendix Figure 9).

For MN, also the benchmarks SO and STM perform better than transfer learning and TO is as bad as transfer learning. This underlines the unsuitability of the data of MN for transfer learning.

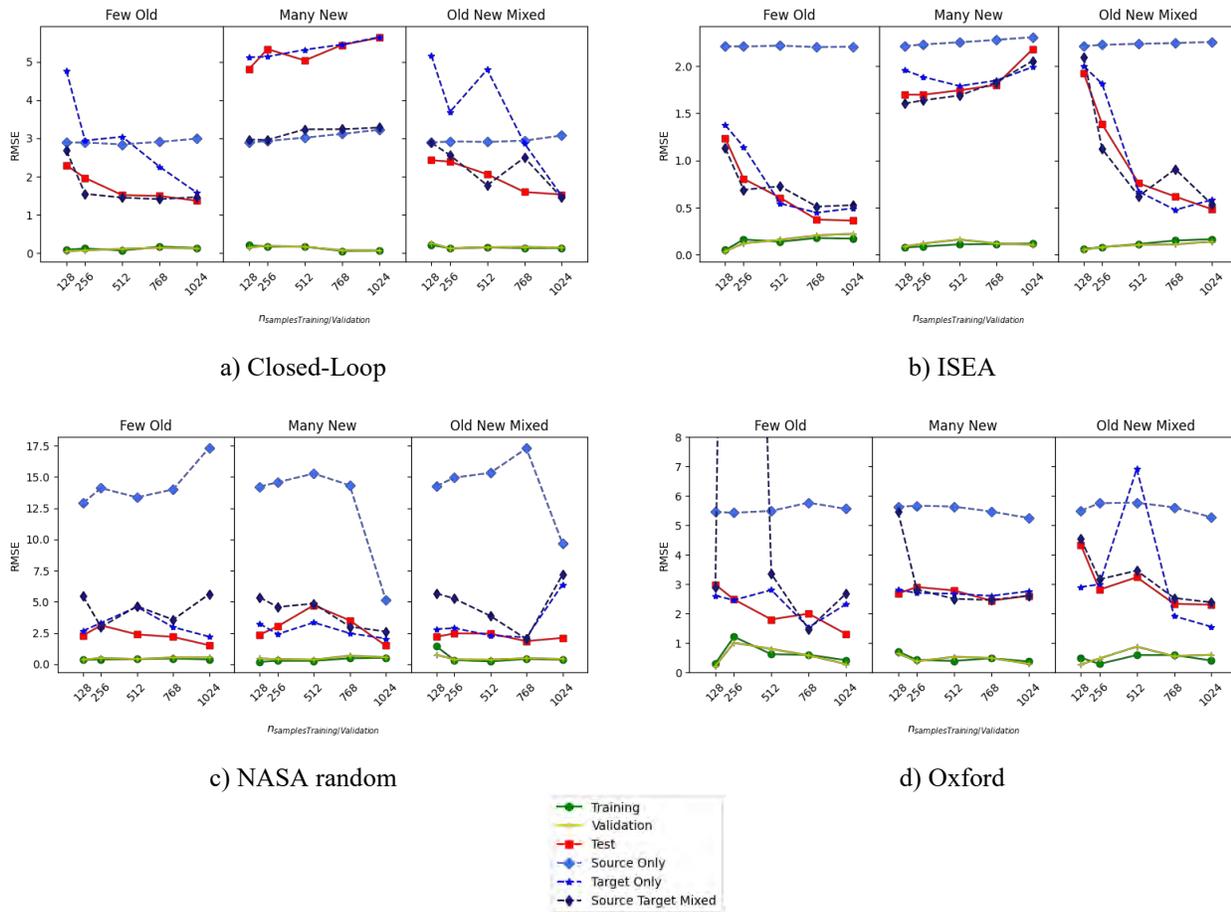


Figure 8: Best models selected by validation RMSE of all four target datasets and their benchmarks

For the ONM, the model performance is similar to FO. Thus adding MN to FO does not provide additional relevant information to the model. Again TO performs worse than the transferred model.

Experiment 2b: ISEA, NASA random, and Oxford

The results using the target datasets ISEA, NASA random, and Oxford confirm that FO is the preferred scenario of data availability. Only for NASA random and Oxford the test RMSE of FO and MN is similar, but still FO is better. We assume that this is caused by a constant ageing rate of NASA random and Oxford (see Appendix Figure 9). Contrarily, DD, CL, and ISEA show superlinear degradation and, thus, have a knee point. This knee point makes forecasting with the MN data split more difficult than when having linear degradation because data of the accelerated ageing after the knee point is missing. Consequently, the test RMSE of ONM of CL and ISEA is better than of MN because ONM contains information about the knee point.

Overall, the SO benchmark is worse than the other benchmarks and transfer learning except in 16% of the cases. This indicates that any data from the target dataset improves the model's performance. The comparison of the TO and STM benchmarks with transfer learning leads to no unambiguous results as each of the three is performing slightly better in some cases. Unfortunately, we could not identify any clear pattern under which circumstances which of the three is preferable. Probably, a more extensive source dataset is required for a clear added value of transfer learning. One could for example, use four of the presented datasets as source dataset and only one as target dataset.

6. CONCLUSION

This paper showed under which conditions transfer learning enables SOH forecasting based on a comprehensive study using four known public target battery datasets: Having data of few aged batteries shall be preferred over data of many young batteries, especially in the case of superlinear degradation with knee points. In the case of data of few aged batteries, having data of more aged batteries from the target domain will improve model performance. We also discussed the problem of an available metric (Validation vs. test RMSE) which prohibits selecting the best possible model.

In contrast to state of the art transfer learning in computer vision, freezing no layers was best for the transfer with most datasets and data splits. This is due to the limited amount of source data and coherent to the results of Shen et al. (2020).

The results are biased by the MLP architecture chosen by a hyperparameter optimization based on the source dataset. Furthermore, none of the dataset originates from BEV operation, but only laboratory data was used. Only the NASA random and Oxford dataset come close to BEV operational data with variability in discharge c-rate and calendar ageing respectively. Thus, transfer learning from laboratory to BEV

operational data is planned as future work. In that case, possibly another advantage of transfer learning might become relevant if models are trained on thousands of samples from BEV fleets. Like in computer vision, then transfer learning might save usage of computational resources when pre-training a model requires weeks of training, but transfer learning significantly less.

In this work, we only considered a single battery type each in the source and target domain. Now that conditions for transfer learning enabling SOH forecasting are known, future work could examine SOH forecasting for several battery types whose data become available sequentially. This paradigm is called continual learning. Continual learning is similar to transfer learning, but aims not only at transferring to a single new task, but to several sequentially occurring tasks. Thus, catastrophic forgetting after the first new task becomes relevant for continual learning (Parisi, Kemker, Part, Kanan, and Wermter 2019). Continual learning in the context of SOH forecasting is imaginable in two scenarios: First, with progressing time more and more data of the same battery type or fleet will be available. Each time, integrating this new data into an existing model is a task of continual learning. Second, a single SOH forecasting model may be desirable for several battery types of different generations that are only developed in temporal sequence. Once, data of a new battery type is accessible an existing model shall be made suitable for the old battery types and the new one.

Connected to continual learning, curricular learning is worth examining to find a training strategy that presents training samples not randomly, but organized in a meaningful order. This enables the model to gradually learn more complex concepts (Bengio, Louradour, Collobert, and Weston 2009).

ACKNOWLEDGEMENT

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. The results, opinions and conclusions expressed in this publication are not necessarily those of Volkswagen Aktiengesellschaft.

REFERENCES

- Attia, P. M., Grover, A., Jin, N., Severson, K. A., Markov, T. M., Liao, Y.-H., Chen, M. H., Cheong, B., Perkins, N., Yang, Z., Herring, P. K., Aykol, M., Harris, S. J., Braatz, R. D., Ermon, S., & Chueh, W. C. (2020). Closed-loop optimization of fast-charging protocols for batteries with machine learning. *Nature*, vol. 578 (7795), pp. 397–402. doi: 10.1038/s41586-020-1994-5
- Attia, P. M., Bills, A. A., Brosa Planella, F., Dechent, P., dos Reis, G., Dubarry, M., Gasper, P., Gilchrist, R., Greenbank, S., Howey, D., Liu, O., Khoo, E., Preger, Y., Soni, A., Sripad, S., Stefanopoulou, A., & Sulzer,

- V. (2022). Review—"Knees" in Lithium-Ion Battery Aging Trajectories. *Journal of The Electrochemical Society*, vol. . doi: 10.1149/1945-7111/ac6d13
- Azkue, M., Lucu, M., Martinez-Laserna, E., & Aizpuru, I. (2021). Calendar Ageing Model for Li-Ion Batteries Using Transfer Learning Methods. *World Electric Vehicle Journal*, vol. 12 (3), p. 145. doi: 10.3390/wevj12030145
- Barré, A., Deguilhem, B., Grolleau, S., Gérard, M., Suard, F., & Riu, D. (2013). A review on lithium-ion battery ageing mechanisms and estimations for automotive applications. *Journal of Power Sources*, vol. 241 (9), pp. 680–689. doi: 10.1016/j.jpowsour.2013.05.040
- Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, vol., pp. 41–48. doi: 10.1145/1553374.1553380
- Bertoldi, N., Cettolo, M., Federico, M., & Buck, C. (2012). Evaluating the Learning Curve of Domain Adaptive Statistical Machine Translation Systems. *Proceedings of the 7th Workshop on Statistical Machine Translation Montréal, Canada, June 7-8, 2012*, vol., pp. 433–441.
- Bole, B., Kulkarni, C. S., & Daigle, M. (2012). *Randomized Battery Usage Data Set*. Available at: <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/> (last accessed February 4, 2021).
- Chen, L., Lü, Z., Lin, W., Li, J., & Pan, H. (2018). A new state-of-health estimation method for lithium-ion batteries through the intrinsic relationship between ohmic internal resistance and capacity. *Measurement*, vol. 116, pp. 586–595. doi: 10.1016/j.measurement.2017.11.016
- Choi, Y., Jung, D., Ham, K., & Bae, S. (2011). A study on the accelerated vibration endurance tests for battery fixing bracket in electrically driven vehicles. *Procedia Engineering*, vol. 10, pp. 851–856. doi: 10.1016/j.proeng.2011.04.140
- Fermin-Cueto, P., McTurk, E., Allerhand, M., Medina-Lopez, E., Anjos, M. F., Sylvester, J., & dos Reis, G. (2020). Identification and machine learning prediction of knee-point and knee-onset in capacity degradation curves of lithium-ion cells. *Energy and AI*, vol. 1 (5), p. 100006. doi: 10.1016/j.egyai.2020.100006
- Gassner, E. (1984). Vereinfachter Betriebsfestigkeits-Nachweis für zufallsartig im hohen Lebensdauerbereich beanspruchte Fahrzeugbauteile / Time-reduced performance fatigue test for automotive components randomly loaded in the high endurance range. *Materials Testing*, vol. 26 (8), pp. 274–276. doi: 10.1515/mt-1984-260807
- Gewald, T., Candussio, A., Wildfeuer, L., Lehmkuhl, D., Hahn, A., & Lienkamp, M. (2020). Accelerated aging characterization of lithium-ion cells. Using sensitivity analysis to identify the stress factors relevant to cyclic aging. *Batteries*, vol. 6 (1), p. 6. doi: 10.3390/batteries6010006
- Keil, P. (2017). *Aging of Lithium-Ion Batteries in Electric Vehicles*. PhD Thesis, Technische Universität München, München, Germany, <https://mediatum.ub.tum.de/doc/1355829/file.pdf>.
- Leuthner, S. (2018). Lithium-ion battery overview, InR. Korthauer (Ed.), *Lithium-Ion Batteries: Basics and Applications* (pp. 13–19). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Lipu, M. S. H., Hannan, M. A., Hussain, A., Hoque, M. M., Ker, P. J., Saad, M. H. M., & Ayob, A. (2018). A review of state of health and remaining useful life estimation methods for lithium-ion battery in electric vehicles. Challenges and recommendations. *Journal of Cleaner Production*, vol. 205, pp. 115–133. doi: 10.1016/j.jclepro.2018.09.065
- Marongiu, A., Roscher, M., & Sauer, D. U. (2015). Influence of the vehicle-to-grid strategy on the aging behavior of lithium battery electric vehicles. *Applied Energy*, vol. 137, pp. 899–912. doi: 10.1016/j.apenergy.2014.06.063
- Matadi, B. P., Geniès, S., Delaille, A., Waldmann, T., Kasper, M., Wohlfahrt-Mehrens, M., Aguesse, F., Bekaert, E., Jiménez-Gordon, I., Daniel, L., Fleury, X., Bardet, M., Martin, J.-F., & Bultel, Y. (2017). Effects of biphenyl polymerization on lithium deposition in commercial graphite/NMC lithium-ion pouch-cells during calendar aging at high temperature. *Journal of The Electrochemical Society*, vol. 164 (6), A1089-A1097. doi: 10.1149/2.0631706jes
- Nuhic, A., Terzimehic, T., Soczka-Guth, T., Buchholz, M., & Dietmayer, K. (2013). Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods. *Journal of Power Sources*, vol. 239 (3), pp. 680–688. doi: 10.1016/j.jpowsour.2012.11.146
- Palmer, J. A. (2003). *Relative Convexity*. Available at: https://scn.ucsd.edu/~jason/relcon_new.pdf (last accessed December 16, 2021).
- Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, vol. 22 (10), pp. 1345–1359. doi: 10.1109/TKDE.2009.191
- Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural

- networks: A review. *Neural Networks*, vol. 113, pp. 54–71. doi: 10.1016/j.neunet.2019.01.012
- Raj, T., Wang, A. A., Monroe, C. W., & Howey, D. A. (2020). Investigation of Path - Dependent Degradation in Lithium - Ion Batteries**. *Batteries & Supercaps*, vol. 3 (12), pp. 1377–1385. doi: 10.1002/batt.202000160
- Rawat, W., & Wang, Z. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, vol. 29 (9), pp. 2352–2449. doi: 10.1162/neco_a_00990
- Razavian, A. S., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 512–519), 2014-06-23/2014-06-28, Columbus, OH, USA. doi: 10.1109/CVPRW.2014.131
- Richardson, R. R., Osborne, M. A., & Howey, D. A. (2019). Battery health prediction under generalized conditions using a Gaussian process transition model. *Journal of Energy Storage*, vol. 23, pp. 320–328. doi: 10.1016/j.est.2019.03.022
- Ruder, S., Peters, M. E., Swayamdipta, S., & Wolf, T. (2019). Transfer Learning in Natural Language Processing. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials* (pp. 15–18), Minneapolis, Minnesota. doi: 10.18653/v1/N19-5004
- Sauer, D. U. (2021). *Time-series cyclic aging data on 48 commercial NMC/graphite Sanyo/Panasonic UR18650E cylindrical cells*. Available at: <https://doi.org/10.18154/RWTH-2021-04545> (last accessed July 12, 2021).
- Severson, K. A., Attia, P. M., Jin, N., Perkins, N., Jiang, B., Yang, Z., Chen, M. H., Aykol, M., Herring, P. K., Fraggedakis, D., Bazant, M. Z., Harris, S. J., Chueh, W. C., & Braatz, R. D. (2019). Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy*, vol. 4 (5), pp. 383–391. doi: 10.1038/s41560-019-0356-8
- Shao, L., Zhu, F., & Li, X. (2015). Transfer learning for visual categorization: a survey. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26 (5), pp. 1019–1034. doi: 10.1109/TNNLS.2014.2330900
- Shen, S., Sadoughi, M., Li, M., Wang, Z., & Hu, C. (2020). Deep convolutional neural networks with ensemble learning and transfer learning for capacity estimation of lithium-ion batteries. *Applied Energy*, vol. 260 (9), p. 114296. doi: 10.1016/j.apenergy.2019.114296
- Simonyan, K., & Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Available at: <http://arxiv.org/pdf/1409.1556v6>.
- Sulzer, V., Mohtat, P., Aitio, A., Lee, S., Yeh, Y. T., Steinbacher, F., Khan, M. U., Lee, J. W., Siegel, J. B., Stefanopoulou, A. G., & Howey, D. A. (2021). The challenge and opportunity of battery lifetime prediction from field data. *Joule*, vol. 5 (8), pp. 1934–1955. doi: 10.1016/j.joule.2021.06.005
- Torrey, L., & Shavlik, J. (2010). Transfer Learning, In E. S. Olivas, J. D. M. Guerrero, M. Martinez-Sober, J. R. Magdalena-Benedito, & A. J. Serrano López (Eds.), *Handbook of Research on Machine Learning Applications and Trends* (242-264). Hershey, PA, USA: IGI Global.
- Viering, T., & Loog, M. (2021). *The Shape of Learning Curves: a Review*. Available at: <http://arxiv.org/pdf/2103.10948v1>.
- von Bülow, F., Heinrich, F., & Meisen, T. (2021). Fleet Management Approach for Manufacturers displayed at the Use Case of Battery Electric Vehicles. *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 3218–3225), 2021-10-17/2021-10-20, Melbourne, Australia. doi: 10.1109/SMC52423.2021.9658680
- von Bülow, F., Mentz, J., & Meisen, T. (2021). State of health forecasting of Lithium-ion batteries applicable in real-world operational conditions. *Journal of Energy Storage*, vol. 44, p. 103439. doi: 10.1016/j.est.2021.103439
- von Bülow, F., & Meisen, T. (2022). A Review on Methods for State of Health Forecasting of Lithium-Ion Batteries applicable in Real-World Operational Conditions. *Unpublished*, vol. .
- von Srbik, M.-T. (2015). *Advanced lithium-ion battery modelling for automotive applications*. PhD Thesis, Imperial College London, London.
- Vuorilehto, K. (2018). Materials and function, In R. Korthauer (Ed.), *Lithium-Ion Batteries: Basics and Applications* (pp. 21–28). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Waag, W., Fleischer, C., & Sauer, D. U. (2014). Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles. *Journal of Power Sources*, vol. 258, pp. 321–339. doi: 10.1016/j.jpowsour.2014.02.064
- Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *27th International Conference on Neural Information*

Processing Systems (NIPS) (pp. 3320–3328), 2014-12-08/2014-12-13, Montreal, Canada.

APPENDIX

Table 5: Signal interval width for current, temperature, and SOC

	Signal interval width		
	I	T	SOC
Fine (F)	0.5C	0.5 °C	5 %
Medium (M)	1C	1 °C	10 % at 0 and 100%, else 20 %
Coarse (C)	3C	3 °C	20 %

Table 4: Combined signals for 2D stressor tables, variant A

variant A	Charging	Discharging	Hold
	T & SOC I & SOC I & T	T & SOC I & SOC I & T	T & SOC

Table 6: Overview of battery operation Oxford dataset

Dataset part	Group no.	Cell no.	Cycling			Calendar aging	
			Duration [d]	C-rate	Cycling type	Duration [d]	SOC _{Storage}
1 & 2	1	9, 15, 20	1	C/2	CC	5	90%
	2	3, 4, 8		C/4			
	3	10, 11, 14	2	C/2		10	
	4	12, 18, 19		C/4			
2	5	5, 6, 16	From 2.5V-4.2V	C/2	CC	-	-
	6	1	-	-	-	-	90%
3	7	4, 19, 27	1	C/2	CCCV	5	90%
	8	5, 18, 20	2			10	90%
	9	1, 8, 9	1			5	4.2V
	10	2, 22, 25				10	4.2V

Table 7: Overview of battery operation NASA random dataset. *CC(CV): CC then CV if enough time

Group no.	Cell no.	Charging			Discharging			$T_{Ambient}$ [°C]
		Duration	C-rate	Cycling type	C-rate	Length random sequence	High probability	
1	9-12	5 min or U=4.2V	Rand. up to 4.5A	CC(CV)*	Rand. up to 4.5A	5 min or U=3.2V	-	20
2	3-6	To 4.2V	2A	CCCV			-	20
3	1,2,7,8	Rand. 0.5 - 3 h	C/2	CC(CV)*	Rand. 0.5-4A	5 min	-	20
4	25-28						High <i>I</i>	40
5	17-20	To 4.2V	2A	CCCV	Rand. 0.5-2A	1 min		20
6	21-24						Low <i>I</i>	40
7	13-16							20

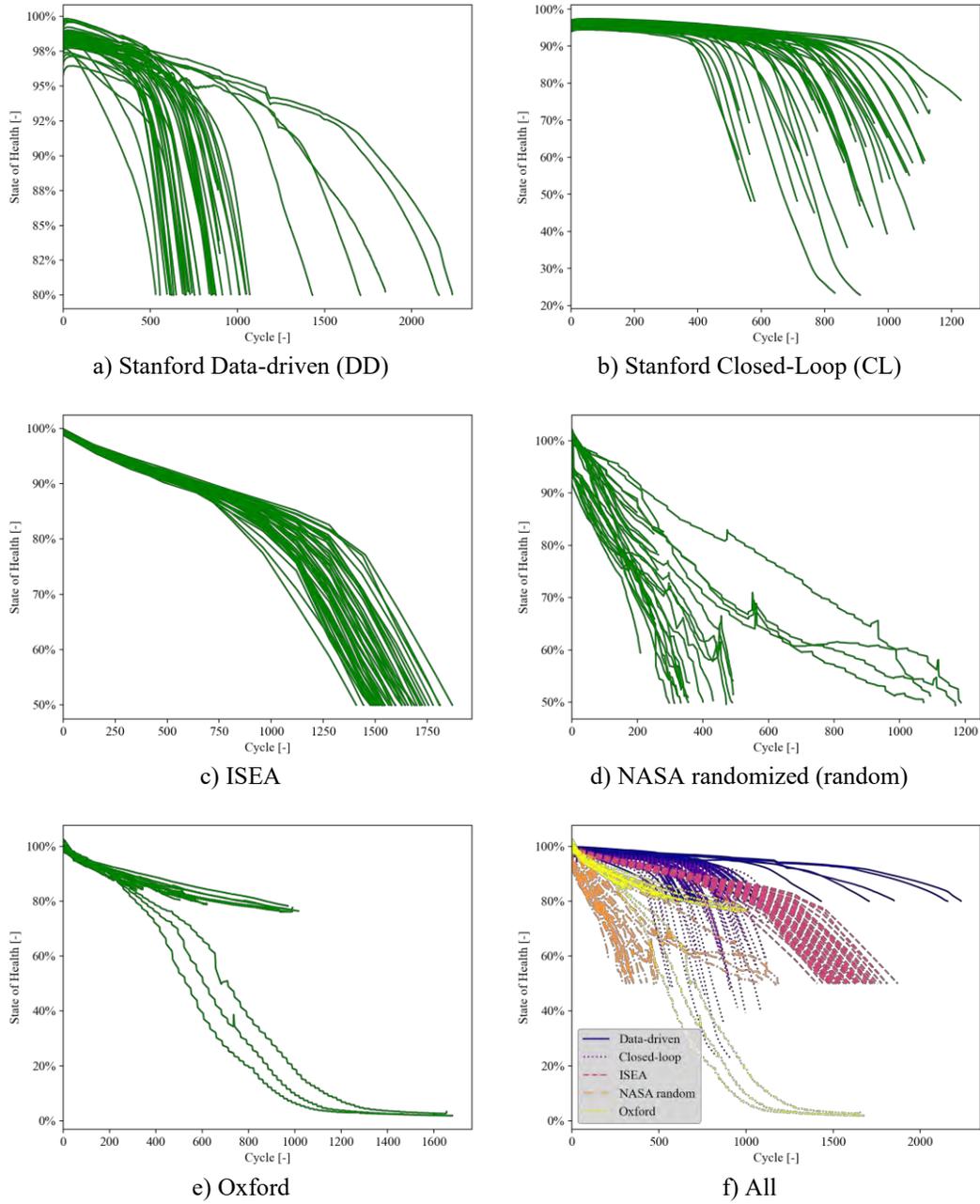


Figure 9: SOH over cycle number k of all used datasets. Discontinuities appear because of plotting over cycles, but the SOH is interpolated over time for ISEA, NASA Random, and Oxford.

Table 8: Hyperparameters of the source model

		{25, 50, 100} cycles (W9)
Hyperparameters		
Activation Function		ReLU
Batch Size		64
Learning Rate α		0.0001
Regularization $\{\lambda_1, \lambda_2\}$		{0, 0.001}
Dropout rate		0
MLP layout		[195, 45, 395, 95, 245, 295, 245, 145, 245, 245]
Model Complexity		
No. of Hidden Layers		10
No. of Model Parameter		1,693,126
Metrics		
RMSE	Train	0.0861
	Validation	0.1032
	Test	0.1083
R²	Train	0.9967
	Validation	0.9961
	Test	0.9935

Failures Mapping for Aircraft Electrical Actuation System Health Management

Chengwei Wang¹, Ip-Shing Fan², and Stephen King³

^{1,2,3}*Integrated Vehicle Health Management Centre (IVHM), School of Aerospace, Transport and Manufacturing (SATM), Cranfield University, Bedford, MK43 0AL, UK*

Chengwei.wang@cranfield.ac.uk

I.S.Fan@cranfield.ac.uk

S.P.King@cranfield.ac.uk

ABSTRACT

This paper presents the different types of failure that may occur in flight control electrical actuation systems. Within an aircraft, actuation systems are essential to deliver physical actions. Large actuators operate the landing gears and small actuators adjust passenger seats. As developing, aircraft systems have become more electrical to reduce the weight and complexity of hydraulic circuits, which could improve fuel efficiency and lower NO_x emissions. Electrical Actuation (EA) are one of those newly electrified systems. It can be categorized into two types, Electro-Hydraulic Actuation (EHA) and Electro-Mechanical Actuation (EMA) systems. Emerging electric and hydrogen fuel aircraft will rely on all-electric actuation. While electrical actuation seems simpler than hydraulic at the systems level, the subsystems and components are more varied and complex. The aim of the overall project is to develop a highly representative Digital Twin (DT) for predictive maintenance of electrical flight control systems. A comprehensive understanding of actuation system failure characteristics is fundamental for effective design and maintenance. This research focuses on the flight control systems including the ailerons, rudders, flaps, spoilers, and related systems. The study uses the Cranfield University Boeing 737 as the basis to elaborate the different types of actuators in the flight control system. The Aircraft Maintenance Manual (AMM) provides a baseline for current maintenance practices, effort, and costs. Equivalent EHA and EMA to replace the 737 systems are evaluated. In this paper, the components and their failure characteristics are elaborated in a matrix. The approach to model these characteristics in DT for aircraft flight control system health management is discussed. This paper contributes to the design, operation and support of aircraft systems.

Chengwei Wang *et al.* This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

The use of Electrical Actuators in aircraft is increasingly adopted in recent years along with the development and use of the “Power-by-Wire” (PbW) technology. On modern aircraft, the number of actuators is on the increase. Small actuators are widely used in aircraft subsystems, such as adjusting passenger seats and controlling the cargo door. Large actuators are used in flight control and landing gear systems.

Electro-Hydraulic Actuator and Electro-Mechanical Actuator technologies have been considered mature and found in service on multiple recent large passenger aircraft types. Compared to fully hydraulic primary and secondary flight control system used in the older Boeing 737s, EMAs and EHAs replace some of the hydraulic actuators in the newer Boeing 787 as shown in Figure 1. EMAs are used for driving mid-spoiler surfaces, and trimmable horizontal stabilizers. In the Airbus 380 (no longer in production) and the new Airbus 350 XWB, EMAs are used for driving secondary flight control (flaps and slats) whereas EHAs are used for driving both primary (aileron, elevator, and rudder) and secondary flight control (spoiler and trimmer) shown in Figure 2 and summarized in Table 1.

As more EA systems are used in aircrafts, the loss of local redundancy prompted urgent concern for updates in the generic maintenance rules to manage these new systems:

- Electrical components will need to be inspected as work tasks in maintenance activities.
- An increase in testing, e.g., high current test is necessary to validate the flexibility of junction installations.
- Procedures will need to change to ensure operators safety during removal & installation.

Table 1. EA systems in flight control on Boeing 787 and Airbus 350 XWB

Aircraft Type	Primary Flight Control				Secondary Flight Control		
	Aileron	Elevator	Rudder	Flaps Slats	Spoiler	Trimmer	
Boeing 787	EHA	EHA		EHA	EHA, EMA	EMA	
Airbus 350 XWB	Hydraulic, EHSA	EHSA		EMA	Hydraulic, EBHA	EHSA	

- Maintenance planning will be impacted when employing preventive maintenance, or scheduled maintenance.

Intelligent predictive maintenance scheduling needs to be developed for better and more economical maintenance without comprising safety redundancy.

This study reviews the development and working principles of EA systems, discusses on failure characteristics at component/subsystem level, and consolidate the base for DT model development. The paper is structured as follows. Section 2 reviews the trend of aircraft electrification and its benefits as a basis. Section 3 discusses the advantages and limitations of EA systems, including EMA and EHA, by comparing with the traditional centralized hydraulic actuation system, and to establish the economic basis of employing EA systems in modern new passenger and cargo aircrafts. Section 4, 5 and 6 describe the structure and configurations of different types of EA systems. The study uses the Cranfield University Boeing 737 as the baseline to elaborate the different types of actuators in the flight control system. In Section 7 and 8, the authors discuss the failure characteristics at component and system levels. Two matrices are used to summarize the interconnections between system, component, failure characteristics and features. Section 9 explains the approach to model these characteristics in DT for aircraft flight control system health management. Section 10 concludes the paper.

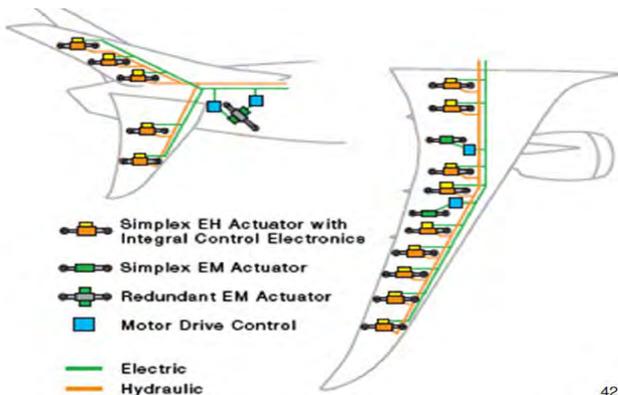


Figure 1. Boeing 787 primary flight control system (MOOG. Servo Hydraulic Technology in Flight Control)

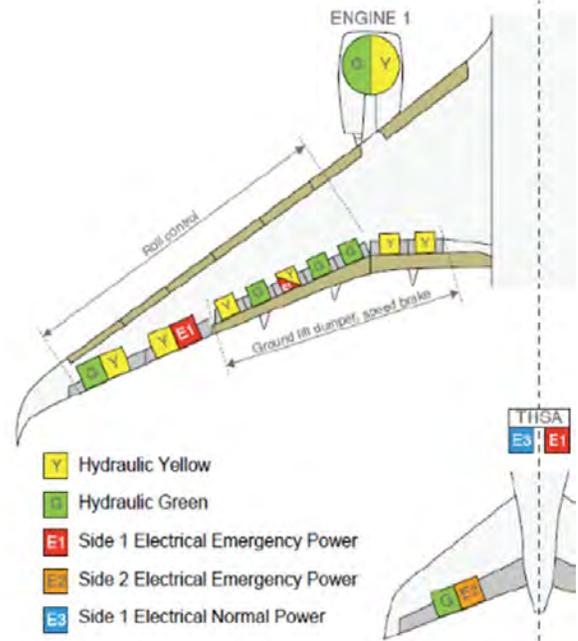


Figure 2. Airbus 350 XWB actuator and power source distribution on left wing, engine, and elevator (AIRBUS. A350 XWB: Flight Control)

2. BACKGROUND

The concern on higher jet fuel prices and operational cost, and the need of aviation sustainability and greenhouse emission reduction (Zaporozhets et al., 2020; Emmanouil, 2020; Janson et al., 2017) have been driving aircraft development. Sustainable Aviation Fuels (SAFs) was developed and used partially in the aviation industry with an advantage of 80% emission reduction during its full lifecycle. With the development of high output density power supply (Chakraborty et al., 2013; Terao et al., 2019), another development is the More Electric Aircraft (MEA) concept. It uses mainly electrical and electric-hybrid systems instead of combinations of secondary power sources, e.g., hydraulic, pneumatic, and mechanical, to realize certain operations.

Aircraft Electrification (AE) has various benefits. It allows for interchangeability and re-configurability as the conventional aircraft systems have reached a so-called “technology saturation” (Chakraborty et al., 2013). There is concern about increasing complexity of traditional systems and inability to design and optimize them (Telford et al.,

Table 2. Comparison between centralized hydraulic actuation and electrical actuation systems

	Advantage	Weakness/Limitation
Centralized Hydraulic Actuation	Stable, mature Time-tested and proven solution High-amount historical experience	Technology saturation Essentially redundance Added weight, complex subsystem
Electrical Actuation	Weight reduction Operation vulnerability Improved efficiency Largely eliminate losses by PbW system Transferable historical experience Simpler, lighter	Overheating/thermal dynamics issue Increased electrical power requirement Power quality concern Electromagnetic interference Mechanical jamming

2012). Aircraft electrification also brings benefits in aircraft weight reduction and power efficiency.

Lower carbon emission is another significant advantage of electrification. Aviation emissions have accelerated in recent years. The development and extension of commercial aviation continue to raise the industry’s contribution to global emissions (Overton, 2019), which reached to 915 million tons of CO₂ in 2019, an increase of 29% since 2013 (Graver et al., 2020). According to the EU’s Flightpath 2050 Program, it sets a goal of decreasing 75% CO₂ emissions per passenger kilometer and a 90% reduction in NO_x emissions. With the use of fuel cell, it is expected to reduce fuel consumption by 1 to 5% and increase its efficiency by 1 to 3%.

AE can be a driver for increasing the sustainability of connectivity within cities. It is also believed to generate lower perceived noise emission, quieter during operation in another word, which benefits urban area and could meet the target of 65% reduction in Flightpath 2050 program.

There are several electric aircrafts that took to the sky. The Finnish Pipistrel Alpha Electro, a two-seat electric plane, took off in August 2018. This is believed to be a milestone for the Finnish aviation to enter a more nature-friendly future. In September 2020, Cranfield University supported ZeroAvia to successfully deliver the world’s first hydrogen fuel cell powered aircraft, which is a big move towards zero-emission aviation.

3. ADVANTAGES AND LIMITATIONS OF EA SYSTEMS

This section detailed discuss advantages and limitations of electrical actuation system compared to traditional centralized hydraulic actuation system. The comparison is summarized in Table 2.

Centralized hydraulic has the advantage of being a time-tested and proven solution whose operational characteristics are well-understood through decades of aeronautical experience. As a result, there is a wealth of historical experience on the centralized hydraulic actuation system.

Nevertheless, it has reached “technology saturation”, the point of diminishing return beyond which further improvements in operational efficiency are increasingly difficult to achieve. The flight control redundancy mandated by civil regulations essentially requires the incorporation of independent hydraulic lines/systems, which add both weight and complexity but can still be at risk from common-cause failures.

3.1. Advantages

Electrical actuation system offers design benefits in terms of reduced weight and operational vulnerability, improved efficiency, etc.

Improved efficiency can be evaluated in the comparison: traditional centralized systems need to remain energized during the whole duration of the flight. In contrast, the “Power-by-Wire” system only provide the exact function required without concerns of excess or reduced power being supplied, largely eliminating the continuous losses that occur within a hydraulic circuit. This was first used in commercial aviation on the Airbus A380; and developed from the “Fly-By-Wire” firstly used in commercial aviation on Boeing 777. Similar electric approach is shown to bring 3% in lower lifecycle costs and a 6% decrease in gross take-off weight. This evolution speeded up the use of more advanced controllers, which was limited in the earlier stages of their implementation in the hydro-mechanical domain (Maré and Fu, 2017).

Significant **weight reduction** is achieved by using for flight and is summarized in Table 3. Relevant implementations can be found on A380. Using electrical signalling allows for 10% reduction of the trim horizontal stabilizer area. On A350, electrical signalling increases the aerodynamic efficiency through differential flap setting, and adaptive dropped hinged flaps (Maré and Fu, 2017).

The reduction in aircraft empty weight directly translates into reduced cost of ownership and operation.

Table 3. Weight reduction and implementation of Airbus aircraft products (Maré and Fu, 2017)

	Model				
	A300-B4	A310	A320	A340	
				A340-200	A340-500/600
Weight reduction	0	-300kg	-200kg	N/A	-50kg*
Implementation	0			-45% of rudder actuator weight**	

*50kg mass reduction is compared to A340-200

**A 45% mass reduction was achieved in A340 series compared to A300.

3.2. Limitations

Regarding the weakness of EA, the thermal dynamics of the actuator units is a crucial consideration since they act as localized sources of heat. The increased electrical quality requirement and the nature of the power draw requires a more elaborate consideration of power quality management and electromagnetic interference. From the design point, there is a disparity in the amount of historical experience available for hydraulic actuators versus electrical ones.

Comparing EHA to EMA, though it is heavier and more complicated, it is more suitable for side-by-side implementation with centralized hydraulics, as demonstrated EBHA units. Some historical experience with components is also transferable to the design of EHAs. The EHA system in the linear modeling method has other disadvantages, such as ignoring feedback loops with certain nonlinearity, and simplifying the friction, especially the static friction. The EMA is simpler and lighter, but it has the risk of mechanical jamming and overheating problems.

4. ELECTRICAL ACTUATION SYSTEM

There are four types of Electrical Actuation systems to replace traditional hydraulic ones.

4.1. Electro-Mechanical Actuation system

The Electro-Mechanical Actuation system is an explicit class of motion control technology systems in which it converts electricity to mechanical force to perform some type of work, e.g., control the speed and/or the position of the end-use, by means of an electric motor and a mechanical transmission, such as a reduction gearbox or a worm screw driver (SAE, 2020). The system generally consists of an Electronic Control Unit (ECU), an Electronic Power Unit (EPU), an electric motor, a mechanical drive and a position sensor.

4.2. Electro-Hydraulic Actuation servo system

The Electro-Hydraulic Actuation servo (EHA) system has electric and hydraulic circuits collaborating to drive a hydraulic actuator. According to MOOG, a typical EHA consists of six major elements (shown as Figure 3): (1)

control electronics (for example, control computer in the cockpit, or guidance system) to create command input signals; (2) servo-amplifier to provide a low power actuating signal, differentiating input signal from feedback signal generated by feedback transducer; (3) servo-valve which responds to actuating signal and controls the flow within the hydraulic circuits to the (4) actuator’s piston or cylinder; (5) power supply, normally an electric motor and pump, delivering high-pressure flow of hydraulic fluid within circuits; (6) a feedback transducer measures the position of the actuator (fully extended or retrieve or in progress) and converts this measurement into a proportional signal. The signal will be fed back to the servo-amplifier.

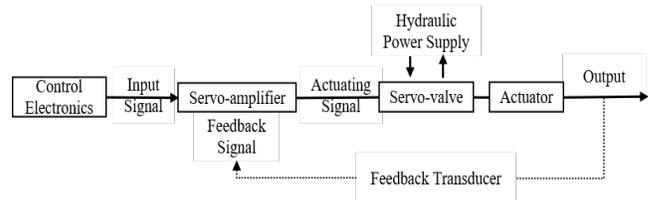


Figure 3. Schematic of a typical EHA (MOOG)

4.3. Electro-Hydro-Static actuation system

The Electro-Hydro-Static Actuation system (EHSA) employs an electric motor to drive a bidirectional hydraulic pump of typically fixed displacement by adjusting its steering and flow output. This completely self-contained system combines electric and electrohydraulic actuation elements. It receives power from an electric source and input a command signal (controls from cockpit via PbW) into motion. A typical EHSA includes a servomotor, hydraulic pump, accumulator, and servo-actuator (MOOG), shown as Figure 4.

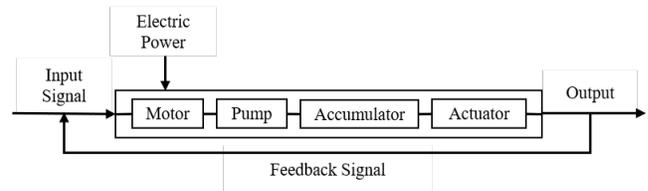


Figure 4. Schematic of a typical EHSA (MOOG)

Differentiating from EHA, it is the servo motor, or a stepping motor, to be controlled after the input signal (target command signal) and feedback signal are amplified.

4.4. Electrical-Backup Hydraulic Actuation system

The Electrical-Backup Hydraulic Actuation system (EBHA) is totally segregated from the normal flight control system and is a combination of a conventional servocontrol and an EHA. In normal mode, it operates as conventional actuator. If there is a hydraulic failure, it operates as EHA.

5. CONFIGURATIONS OF EA SYSTEMS

This section discusses on several working configurations of EMAs and EHSAs.

5.1. EMA

In general, an EMA (shown as Figure 5, also mentioned in Section 4.1) is comprised of an Electronic Control Unit (ECU), an Electronic Power Unit (EPU), an Electric Motor (EM), a Mechanical Drive (MD) and a position sensor.

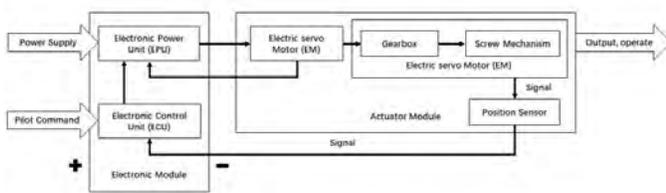


Figure 5. Schematic of a gear-driven EMA (Qiao et al., 2017)

5.2. EHSA

According to the controlling mode of motor and pump, EHSA can be categorized into three types: Variable Pump and Fixed Motor (EHSA-VPFM), Fixed Pump and Variable Motor (EHSA-FPVM), Variable Pump and Variable Motor (EHSA-VPVM).

5.2.1. EHSA-VPFM

The EHSA-VPFM (shown as Figure 6) employs a fixed-speed motor and a servo motor to control the swash plate angle of the axial piston pump and pump displacement. (Alle, Hiremath, Makaram, Subramanian, and Talukdar, 2016).

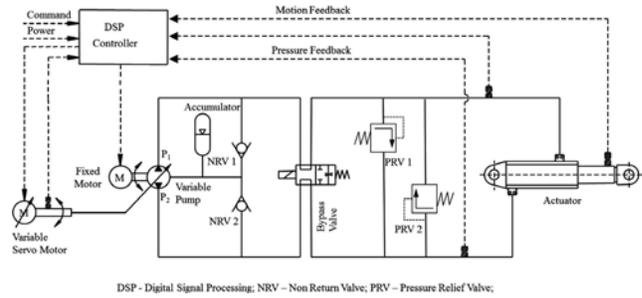


Figure 6. Schematic of an EHSA-VPFM (Alle et al., 2016)

5.2.2. EHSA-FPVM

The EHSA-FPVM (shown as Figure 7) has slower dynamic response than EHSA-VPFM, but its efficiency is relatively higher (Fu, Yang, Qi, Zhang, 2011), and it benefits from a simple structure. In this system, a bi-directional pump rotates at variable speeds and directions driven by an electric motor. As a result, the oil flow and supply pressure are variable to drive the symmetrical actuator.

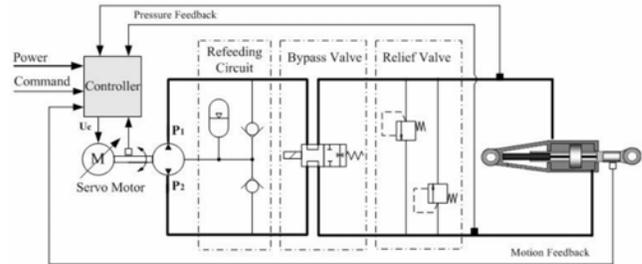


Figure 7. Schematic of an EHSA-FPVM

Kang et al. (2008) proposed a block diagram modeling method of the EHSA-FPVM to address concerns mentioned in Section 3.2. A nonlinear accuracy model is established by block diagrams, which contains more information than conventional linear model. The comparison analysis indicates the effect of EHSA refeeding circuit on reducing the pressure ripple. A gain-variable PID controller is introduced and efficiently compensates the friction. Huang et al. (2020) researched on a novel configuration, Active Load-Sensitive Electro-Hydrostatic Actuator (ALS-EHSA), by adding an active load sensing circuit, which consists of a pressure follow servo valve and a shuttle valve, on the EHSA-FPVM system. The proposed ALS-EHSA can reduce motor heating. Thanks to higher impedance, it reaches smaller displacement tracking error, near zero speed.

5.2.3. EHSA-VPVM

The structure of an EHSA-VPVM is shown as Figure 8. It consists of bi-directional variable pump, controller sensors, hydraulic system, and two servo motors that adjust the displacement of pump and the rotational speed of the pump respectively.

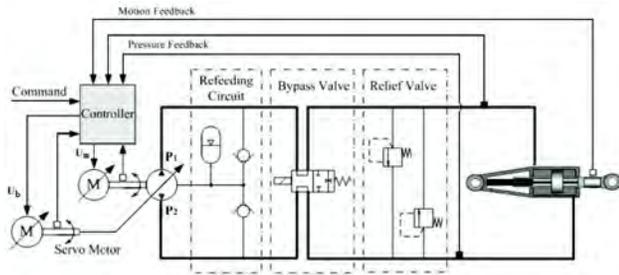


Figure 8. Schematic of an EHSA-VPVM

6. COMPONENTS OF EA SYSTEMS

In a complex EA, multiple components collaboratively operate to realize certain required commands from the pilot, e.g., the extension and retraction of flaps, controlling primary flight control surface. In another word, proper operation of these functions depends on stable performance of the components. Random faults happening in one of components could cause failure to the whole system.

Therefore, this section introduces and discusses the main components involved in the different types of EAs. Moreover, this section builds a foundation for evaluating the failure characteristics of systems in the next section.

6.1. Electrical motor

Considering the overall volume and mass, as well as long in-flight operation of an aircraft, a compact motor with high reliability and high-power density is required in flight control. In an electric aircraft, the speed of the motor would be in the range of 16000-24000 rpm or sometimes even higher in order to increase the power density. The choice of motors normally depends on the power supply on board. Common motor options are permanent magnet synchronous motor (PMSM), brushless DC (BLDC) motor, and switched reluctance (SR) motor. For example, the PMSM motor can be a suitable choice for its advantages in high performance, higher power density than induction motors in same ratings range, and fast acceleration and deceleration. While BLDCs are more cost-effective.

6.2. Gearbox

If the output of an EMA is rotary, the last component of the kinematic chain will be a gearbox, which is interposed between the motor and the screw providing speed reduction. The main purpose of the gearbox is to decelerate the low-torque servo motor and drive a screw mechanism to reach low speed and high torque. Harmonic gear reducers or cycloidal reducers is commonly used due to its compact structure, ease to achieve zero backlash, high reduction ratio and efficiency.

6.3. Screw Mechanism

If the output of an EMA is linear the last component of the kinematic chain will be a power screw, most of the time featuring the rolling element. The motor shaft is directly connected to the screw. This mechanism is employed to transform the rotary motion to linear motion with a required force. Overall weight of the actuator tends to reduce when the transmission ratio increases (Liscouët et al., 2008). While higher transmission ratio can cause a decrease in efficiency and imply a higher rotating input speed. Therefore, the balance between the reducer transmission ratio and the lead of the screw mechanism needs to be considered.

6.4. ECU

As mentioned in Section 4.1 and 4.3, each EA system has a suitable ECU controller, working at 28 Vdc power from external power supply. The ECU receives input command signal from cockpit and transfer to the next processes, including servo motors and EPU. For example, specific voltage input and torque output will be requested by the ECU and been transferred to the motor. Additional temperature sensors are usually installed on the controller housing close to the connectors for flight conditions with high external load. This aims at recording the external temperature changes and trigger alarm at high temperature (Qiao et al. 2017).

6.5. Hydraulic pump

Hydraulic pump in the EHSA is used to convert the torque input received form the motor into hydrostatic pressure and flow in the hydraulic circuits. Referring to Section 5.2, different types of pumps are used in different configurations of EHSA. Gear pumps is exclusive to the fixed displacement type, axial piston pumps and vane pumps can be either of the fixed displacement type or of the variable displacement type.

6.6. Hydraulic actuator

Hydraulic actuator, the operating element of an EHSA, can covert hydraulic pressure and flow into force and velocity of the actuator (Alle et al., 2016). For example, on Airbus 350 XWB, hydraulic actuators of EHSA are used to transfer linear motion and power to extend or retrieve spoilers, flaps, and slats in flight control.

7. FAILURE CHARACTERISTICS OF COMPONENTS

This section focuses on different failures that may occur during operations and consequences impact the subsystems.

7.1. Overheating

Operating temperature is always a concern on mechanism operation. Considering the transmission mechanism responds to commands and only operates for a short period, during which the heat generated from mechanical interaction itself,

e.g., gear meshing, barely attributes overheating issue and then it can be eliminated. Overheating is mainly generated within the electrical servo motor.

There are several reasons for overheating:

Voltage unbalance can produce serious overheating in motors due to the high negative sequence current which flows with a relatively small out-of-balance component. Normally, the motor operates with rated voltage applied and provide rated power. Transient response to pilot's commands and emergency when the voltage applied exceeds its rating. It causes sharp increase in the core magnetic flux density and iron loss and causes the motor to overheat.

Overload is caused by an increase in the load driven by the motor or mechanical jamming in the driven mechanism. Also, mechanical failure happens when rotor and stator are rubbed together, and the rotor is stuck in position.

The **lack of lubricating** oil or grease causes sleeve bearing damage. Electric motors use either rolling or sleeve bearings of lubricated inner and outer ring metal surfaces to reduce friction. These balls or rollers hold the load and support the motor shaft so that the rotor can rotate smoothly and stably. It is normal to choose bearings which will give a life of 50,000-100,000h running, or even 200,000h on the larger motors. Hence, it is essential to ensure minimum wear in normal operation and achieve sufficient use life. Worn shaft collars could failed to isolate the inner bearing from the outside air. Therefore, air will be absorbed in and deteriorate the lubricant grease, corrode balls or rollers. This issue leads to power fluctuation, and uneven and unstable rotor operation as a result, with excessive heat generated.

7.2. Mechanical jamming

Mechanical jamming happens in both motors and transmission mechanisms. As stated in Section 7.1, the rotor rubbed with the stator causes failure of overheating. In the gearbox, worm gear and bearings lead to unstable axial rotation and vibration of transmission shaft. Deteriorated or contamination-collected lubricant grease accelerates tooth wear. Normal failures happen under the above circumstances include gear teeth fracture, teeth surface fatigue pitting, surface glued and plastic deformation.

7.3. Hydraulic leakage

Hydraulic cylinder is the actuator in the hydraulic system, and its failure directly affects the normal operation of the system. The **leakage** of hydraulic cylinder is a major failure mode usually caused by failure or damage of the seal. Air will be absorbed due to worn piston seal or cylinder barrel, which can accelerate wear and tear, excessive internal leakage, and cause **pressure loss** and piston retrieve failure. It should be detected as early as possible to avoid further breakdowns of the system.

7.4. Electric cable wear

The extreme environment these devices operating in, e.g., temperature variation and variable pressure levels, can accelerate **cable wear** in an electric circuit. This may lead to short circuit, higher resistance, overheating or fracture, and loss control in the end.

The control cables in the wing and nacelle area are near high temperature sources. Deterioration of lubricants will occur at a faster rate than on other control cables.

7.5. Unstable power supply

Power generation breakdown is another hidden danger. In the circumstance that one or more of the aircraft's power source fails, and the remaining sources cannot provide sufficient power to maintain the whole system in normal operation until emergency landing would lead to a catastrophe. The constraint frequently imposed by the power supply is the maximum current or kilovolt-amperes which may be drawn during starting, a condition which may be met either by a lower starting current design of motor or by use of some form of soft-start device to reduce the current drawn during starting. When considering the supply constraints, the source impedance must also be considered to ensure that there is sufficient voltage at the machine terminals to overcome the load torque, leaving sufficient torque in hand to accelerate the motor against load.

8. FAILURE FEATURES

Features such as input voltage, output torque, temperature can simultaneously reflect a system's operational status. Failure identification involves selecting features from relevant aspects and filtering unclear features or interference. Further data analysis against parameter uncertainties improves failure identification.

8.1. Features at component level

Correlating components in Section 6 and failure characteristics in Section 7, features can be linked and correlated, as shown in Table 4. For example, bearings that are corroded or lack of lubrication cannot support rotors rotating in a stable condition, irregular vibration could be detected. The frequency of vibration and the vibration of the rotor or the motor can be set as feature references. The two values of frequency and amplitude of normal operation, natural operational condition, can be used as thresholds. When abnormal conditions occur, the feature signals will activate the trigger, which can be a clue of failure in bearings and the specific components involved.

8.2. Failure effects at subsystem level

In Table 5, the fault of systems is traced to specific ranges of failure characteristics at the component level. When a fault alarm shows up in EMA, indicating an abnormal operation, it

Table 4. A summary of fault, related features and measurement reference of components of EA systems

Subsystems/Components	Fault	Feature	Unit
Electric Motor	Electrical overload	Excessive voltage input	+ΔV
		Excessive torque output	+ΔN·m
Mechanism Transmission (Gearbox, screw mechanism)	Overheating	Temperature	+Δ°C
	Irregular vibration	Frequency	(times)
		Amplitude	+Δmm
ECU	Fluid leakage	Pressure drops	+ΔPa
		Shortened displacement	-Δx
Hydraulic Pump	Jamming	Delayed response	+Δt
	Electric cable wear	Irregular torque output	+ΔNM
Hydraulic Actuator		Unstable power supply	Excessive voltage
	Short circuit (Excessive amplitude)		+ΔA
	Excessive resistance		+ΔΩ
		Voltage input drop	-ΔV
		Amplitude input drop	-ΔA
		Insufficient torque output	-ΔN·m

may be caused by electric overload, overheating, irregular vibration (normally caused by worn bearings), gear jamming or unstable power supply, or even worn electric cable.

If a temperature rise is reported in the flight control system, troubleshooting through tracing back to the component level can help to distinguish whether the cause is the motor or the ECU, or either mechanism or hydraulic mechanisms.

This matrix forms the basis for further research on DT modelling, failure simulation, and efficient and sufficient fault diagnosis and locating.

9. HEALTH MONITORING METHODS

Since the development of EA, several configurations of it have been used in recent commercial aircrafts. This section discusses current maintenance methods captured from the

Aircraft Maintenance Manual as well as state-of-the art techniques to be applied to future aircraft maintenance activities.

9.1. Current method during maintenance

According to the Aircraft Maintenance Manual of the Cranfield University Boeing 737, the MSG-3 based preventive/scheduled maintenance program is practiced. Using the inspection of the spoiler part in flight control system as an example (AMM TASK 27-61-00). Current maintenance schedule is divided into two parts, covering in different maintenance checks level. One is to thoroughly inspect the system, including the spoiler control cables, the spoiler mixer and the spoiler actuator (TASK number 27-61-00-211-801). Another one is to carry out general visual

Table 5. A summary of interconnection from system level to component level and their fault characteristics

EA System	Component	Fault
EMA	Gearbox	Electric motor
	Screw mechanism	
EHSA	-VPVM	ECU
	-FPVM	Hydraulic Pump
	-VPFM	Hydraulic actuator
		Electrical overload
		Overheating
		Irregular vibration
		Fluid leakage
		Jamming
		Electric cable wear
		Unstable power supply

inspection on the flight spoiler actuator from the ground (TASK 27-61-00-210-801).

In the thorough inspection TASK 27-61-00-211-801, certain aircraft areas are checked, including left and right main wheel well, both trailing edges outboard and inboard of flap and spoiler, and fixed trailing edges between them, on both left and right wings. The maintenance technician will access the control panels to operate certain surfaces prior, during and after the inspection. The AMM also stated that only corrosion-preventive grease and lint-free cloth are permitted.

Following with a ground visual inspection TASK 27-61-00-210-801, maintenance technicians will do a general visual inspection of the ground and flight spoiler actuators looking for security of installation, leaks, corrosion, and obvious damages. After completing above subtasks, together with other preventive maintenance activity, the maintenance technician can sign-off the aircraft to be ready to go.

Current preventive maintenance reserves sufficient safety redundancy as a system/subsystem will be inspected within a certain flight hour interval according to the check level. A component will be replaced when either found faulty or reaches its pre-set use life. However, in practice, replaced components usually have remaining life, depending on the component, operational environment and human (pilot, maintenance technician) operation behaviours. Therefore, it is more economical and sustainable to seek a better maintenance program to maximize the usage of component/subsystem's life without sacrificing the safety and overall operational flight time as well as turnaround time.

9.2. Potential method

Predictive maintenance can be a potential solution to efficiently monitor EA health. Such maintenance activity is scheduled based on operations of facilities and systems themselves. It employs the Internet of Things (IoT), Big Data and further DT technology. DT is a software design pattern (Gartner, 2019) that presents a physical object to gain a clearer picture of real-world performance and duplicate operations of an asset in real-time referring to real-time collected data from the asset and deduce reliable operation decisions (Qi and Park, 2020), e.g., responding to changes in time and adding value.

9.2.1. Industrial employment

Rolls-Royce released its world-largest aero-engine in early 2021. Referring to the company's IntelligentEngine vision, a DT model is built for each fan blade to store real-life test data, allowing engineers to predict in-service performance.

Air France Industries, KLM Engineering & Maintenance Adaptiveness (AFI KLM E&M) developed the Prognos system, a Big Data-based predictive maintenance solution, which is adapted to multiple aircraft systems and relevant

operations, including Engine, EPCOR for APU, fleet MRO scheduling and inventory optimization.

9.2.2. Research development

Ezhilarasu et al. (2019) investigated the feasibility of DT being combined with Integrated Vehicle Health Management (IVHM) as a decision tool. One of the key roles it can play is to form a platform for integrating information to monitor operations and then duplicate the health status of the complex systems like an aircraft.

Xu et al. (2020) proposed a DT-driven analysis framework (DTAF) for optimizing gas exchange system of 2-stroke heavy fuel aircraft engine. In this research, different DT modules interact with their targeted physical entities of engine and work collaboratively within the DT virtual group. With continuous interaction and correction of DT modules, and real-time data exchange between physical test platforms, DTAF is proven to be efficient and reliable.

Alvarez et al. (2019) developed an DT-corporate solution to aircraft loss of control caused by incorrect measurement readings of the Pitot tube and Air Data Computer (ADC) computing false. This solution aims at using real-time data from other airborne sensors and the virtual sensor DT represented to correctly estimate true airspeed in real-time. The study shows the additional DT virtual sensor can increase the accuracy of the estimation even during downward of the velocity and the altitude.

9.3. Comparison between stated methods

As a mature method, preventive maintenance is well established and regulated alongside the development of aircraft. It benefits from abundant historical documentation, long service experience, and well-trained technicians. Though, regular repetitive activities may lead to human errors. In the crash of Alaska Airlines Flight 261 (NTSB AAR, 2000), the jammed longitudinal trim control system was caused by missed maintenance activities, including no effective lubrication, excessive and accelerated jackscrew wear. The human error is a symptom of ineffective task-by-task engineering analysis and justification.

Predictive maintenance could avoid unnecessary downtime caused by redundant interventions (Zhen et al., 2018). It also prevents losses caused by cascading failures which are affected by untimely maintenance actions. In another word, predictive maintenance helps to increase system reliability while brings down its cost. However, adding more smart techniques and sensors to a certain system adds cost. Predictive maintenance also introduces unplanned dynamics to maintenance demand scheduling and a challenge to the economics of practical maintenance planning.

10. CONCLUSION

This paper discusses in detail EA systems used in aircraft flight control system, including the origin, working principles, subsystem configurations and environmental constraints:

- 1) Background investigation shows AE has become established practice, and EA systems have generally been mature and achieved extensive application on commercial aircrafts.
- 2) A comparison of traditional centralized hydraulic actuation is conducted through access to the AMM of the Cranfield University Boeing 737. The comparison between centralized hydraulic actuation system and EA system is presented to demonstrate the advantages of EA systems, also pointing out key concerns regarding designing and maintenance.
- 3) Different types of EA systems are described, which includes the EMA, and three configurations of EHSAs.
- 4) A detailed analysis is carried out on EA systems by breaking down to the component level, including electrical motor, transmission mechanism, hydraulic pump and cylinder. Failure characteristics of components are evaluated.
- 5) A matrix helping to trace failure in system level via subsystem/component level is presented. It relates the failure characteristics and helps define outstanding features that can be used to indicate correlated faults in further studies.
- 6) Maintenance methods including those carried out in current maintenance activities and novel technologies which are shown in the trend in research are briefly reviewed.

The authors are building on this work to develop a data-driven DT model to represent real EA systems. The validated DT model will be used to support predictive maintenance for EA system health management.

REFERENCE

MOOG. *Primary Flight Control Actuation System for 787*. Available at: <https://www.moog.com/products/actuation-systems/aircraft/primary-flight-control-actuation-system-for-787.html> (accessed: 10 Dec 2020)

AIRBUS. *A350 XWB: Flight Control, Flight Deck and Systems Briefing for Pilots*, p. 8-9. Available at: <https://www.parlonsaviation.com/wp-content/uploads/2017/12/a350-flight-controls.pdf> (accessed: 10 Dec 2020)

Y. Terao, A. Seta, H. Ohsaki, H. Oyori, N. Morioka. (2019). *Lightweight Design of Fully Superconducting Motors for*

Electrical Aircraft Propulsion Systems, IEEE Transaction on Applied Superconductivity, Vol. 29, No. 5, Aug 2019. DOI: 10.1109/TASC.2019.2902323.

Chakraborty, D. Jackson, D. Trawick. (2013). *Development of a Sizing and Analysis Tool for Electrohydrostatic and Electromechanical Actuators for the More Electric Aircraft*, 2013 Aviation Technology, Integration, and Operations Conference, AIAA AVIATION Forum, Aug 2013. DOI: 10.2514/6.2013-4282.

Zaporozhets, V. Isaenko, K. Synylo. (2020). *Trends on current and forecasted aircraft hybrid electric architectures and their impact on environment*, Energy, Vol. 211, Nov 2020. DOI: 10.1016/j.energy.2020.118814.

K. Emmanouil. (2020). *Reliability in the era of electrification in aviation: A systems approach*, Microelectronics Reliability, Vol. 114, Nov 2020. DOI: 10.1016/j.microrel.2020.113945.

R. H. Janson, C. Bowman, A. Jankovsky, R. Dyson, J. Felder. (2017). *Overview of NASA Electrified Aircraft Propulsion Research for Large Subsonic Transports*, 53rd AIAA/SAE/ASEE Joint Propulsion Conference, AIAA Propulsion and Energy Forum, Jul 2017. DOI: 10.2514/6.2017-1701.

IATA. *Developing Sustainable Aviation Fuel (SAF)*. Available at: <https://www.iata.org/en/programs/environment/sustainable-aviation-fuels/#tab-2> (accessed: 10 Dec 2020)

R. D. Telford, S. J. Galloway, G.M. Burt. (2012). *Evaluating the Reliability & Availability of More-Electric Aircraft Power Systems*, 2012 47th IUPEC, Sep. 2012. DOI: 10.1109/IUPEC.2012.6398542.

P. Traverse, I. Lacaze, J. Souyris. (2004). *Airbus Fly-by-Wire: A Total Approach to Dependability*, IFIP International Federation for Information Processing, Vol. 156. DOI: 10.1007/978-1-4020-8157-6_18.

G.F. Bartley. (2001). *Boeing B-777: Fly-By-Wire Flight Controls*, Chap. 11, the Avionics Handbook, Boca Raton, CRC Press. Available at: https://www.davi.ws/avionics/TheAvionicsHandbook_Contents.pdf (accessed: 10 Dec 2020)

J. W. Ramsey. (2001). *Power-by-Wire*, Aviation Today. Available at: <https://www.aviationtoday.com/2001/05/01/power-by-wire/> (accessed: 10 Dec 2020)

J. Overton. (2019). *Fact Sheet: The Growth in Greenhouse Gas Emissions for Commercial Aviation*, Environmental and Energy Study Institute, Oct 2019. Available at: <https://www.eesi.org/papers/view/fact-sheet-the-growth-in-greenhouse-gas-emissions-from-commercial-aviation> (accessed: 10 Dec 2020)

ATAG. *Facts & Figures*. Available at: <https://www.atag.org/facts-figures.html> (accessed: 10 Dec 2020)

B. Graver, D. Rutherford, S. Zheng. (2020). *CO2 Emissions from commercial aviation: 2013, 2018, and 2019*, the

- International Council on Clean Transportation, Oct 2020.
- European Commission. *Flightpath 2050: Europe's Vision for Aviation*. Available at: <https://ec.europa.eu/transport/sites/transport/files/modes/air/doc/flightpath2050.pdf> (accessed: 10 Dec 2020)
- IATA. *Aircraft Technology Roadmap to 2050*. Available at: <https://www.iata.org/contentassets/8d19e716636a47c184e7221c77563c93/technology20roadmap20to20205020no20foreword.pdf> (accessed: 10 Dec 2020)
- Pipistrel. *First flight of an electric aircraft in Finland*. Available at: <https://www.pipistrel-aircraft.com/first-flight-of-an-electric-aircraft-in-finland/> (accessed: 10 Dec 2020)
- Cranfield University. *Cranfield supports ZeroAvia's world first hydrogen-electric passenger aircraft flight*. Available at: <https://www.cranfield.ac.uk/press/news-2020/cranfield-supports-zeroavias-world-first-hydrogen-electric-passenger-aircraft-flight> (accessed: 10 Dec 2020)
- SAE. *AIR8012: Prognostics and Health Management Guidelines for Electro-Mechanical Actuators*, SAE International. DOI: 10.4271/AIR8012.
- MOOG. Electrohydrostatic. Available at: <https://www.moog.com/products/actuators-servoactuators/actuation-technologies/electrohydrostatic.html> (accessed: 20 Feb 2022)
- MOOG. Electrohydraulic. Available at: <https://www.moog.com/products/actuators-servoactuators/actuation-technologies/electrohydraulic/> (accessed: 20 Feb 2022)
- J. Maré, J. Fu (2017). *Review on signal-by-wire and power-by-wire actuation for more electric aircraft*, Chinese Journal of Aeronautics, 2017, 30(3), pp. 857-870. DOI: 10.1016/j.cja.2017.03.013
- G. Qiao, G. Liu, Z. Shi, Y. Wang, S. Ma, and T. C. Lim (2017). *A review of electromechanical actuators for More/All Electric aircraft systems*, Journal of Mechanical Engineering Science, 2018, Vol. 232 (22), pp. 4128-4151.
- N. Alle, S. S. Hiremath, S. Makaram, K. Subramaniam and A. Talukdar (2016). *Review on electro hydrostatic actuator for flight control*, International Journal of Fluid Power, 2016, Vol. 17, NO. 2, pp. 125-145.
- Y. Fu, B. Yang, H. Qi, Y. Zhang (2011). *Optimization of the Control Strategy for EHA-VPVM System*, Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, 2011, pp. 1856-1859. DOI: 10.1109/EMEIT.2011.6023001
- R. Kang, Z. Jiao, S. Wu, Y. Shang, J. Maré. (2008). *The Nonlinear Accuracy Model of Electro-Hydrostatic Actuator, 2008 IEEE Conference on Robotics, Automation and Mechatronics*, 2008, pp. 107-111, DOI: 10.1109/RAMECH.2008.4681367.
- L. Huang, T. Yu, Z. Jiao, Y. Li. (2020). *Active Load-Sensitive Electro-Hydrostatic Actuator for More Electric Aircraft*, Appl. Sci. 2020, 10(19), 6978. DOI: 10.3390/app10196978.
- J. Liscouët, J. Maré, S. Orioux (2008). *Automated Generation, Selection and Evaluation of Architectures for Electromechanical Actuators*, 26th International Congress of the Aeronautical Sciences, 2008, pp. 14-19.
- Gartner. *Gartner Survey Reveals Digital Twins Are Entering Mainstream Use*. Available at: <https://www.gartner.com/en/newsroom/press-releases/2019-02-20-gartner-survey-reveals-digital-twins-are-entering-mai> (accessed: 10 Dec 2020)
- B. Qi, H.S. Park. (2020). *Data-driven digital twin model for predicting grinding force*, IOP Conf. Ser.: Mater. Sci. Eng. 916 012092. DOI: 10.1088/1757-899X/916/1/012092.
- Rolls-Royce, (2019). *How Digital Twin technology can enhance Aviation*. Our stories - How Digital Twin technology can enhance Aviation –Rolls-Royce (Accessed: 9 May 2021)
- AFI KLM E&M. *Prognos*. Available at: The MRO Lab - PROGNOS® - Predictive Maintenance (afiklmem.com) (accessed: 19 May 2021)
- C.M. Ezhilarasu, Z. Skaf, I.K. Jennions. (2019). *Understanding the role of a Digital Twin in Integrated Vehicle Health Management (IVHM)*, 2019 IEEE Int. Conf. SMC, Oct 2019. DOI: 10.1109/SMC.2019.8914244.
- Z. Xu, F. Ji, S. Ding, Y. Zhao, Y. Zhou, Q. Zhang, F. Du. (2020). *Digital twin-driven optimisation of gas exchange system of 2-stroke heavy fuel aircraft engine*, Journal of manufacturing Systems. DOI: 10.1016/j.jmsy.2020.08.002.
- O.H. Alvarez, L.B.G. Zea, C. Bil, M.L. Fravolini, M. Napolitano. (2019). *Digital Twin Concept for Aircraft System Failure Detection and correction*, AIAA Aviation Forum, June 2019. DOI: 10.2514/6.2019-2887.
- National Transportation Safety Board, Washington D.C. (2000). *Aircraft Accident Report: Loss of Control and Impact with Pacific Ocean Alaska Airlines Flight 261. McDonnell Douglas MD-83, N963AS. About 2.7 Miles North of Anacapa Island, California*. January 31, 2000, NTSB/AAR-02/01. Available at: <https://www.nts.gov/investigations/AccidentReports/Reports/AAR0201.pdf> (accessed: 12 March 2022)
- S. Zhen, Z. Wu, K. Xu, Q. Wang, P. Han, H. Yang, Y. Chen. (2018). *Predictive Maintenance White Paper*, Shanghai Industrial Technology Institute. Available at: <https://www.siti.sh.cn/> (accessed: 12 March 2022)

BIOGRAPHIE



Chengwei Wang is currently a PhD research student at Cranfield University, Bedfordshire, UK. He was born in Yancheng, China in 1996. He obtained his dual bachelor's degree with First Class Honours in Mechanical Engineering Technology from The University of Greenwich, UK, and the Yancheng Institute of Technology, China in 2018. He received his master's degree

in Aerospace Manufacturing from Cranfield University in 2019. He previously participated as a research assistant in the Natural Resources Institute, University of Greenwich for a 3-month internship. His research focuses on understanding failure characteristics of electrical actuation systems on aircrafts and the development of applied Digital Twin techniques for aircraft system health management. He is also interested in bringing digital techniques into the aviation sector.



Ip-Shing Fan is currently on the Education and Scholarship pathway. He was born and studied in Hong Kong, graduated with First Class Honours in Industrial Engineering. He completed his graduate engineer training at Qualidux Industrial Co Ltd in Hong Kong. He was awarded the Commonwealth Scholarship and completed his PhD in Computer

Integrated Manufacturing in Cranfield. After returning to Hong Kong, he worked as CAD/CAM Manager in Qualidux Industrial Co Ltd, responsible for the introduction of CAD, CAM, and CNC in plastic injection design and engineering. In 1990, Fan started to work in The CIM Institute, endowed by IBM in Cranfield, to carry out research, education, and consultancy in new applications of computers in manufacturing. He led many European and UK funded research programs to create new tools and methods in

knowledge-based engineering design, business performance, quality management, supply chain, and complexity science. He has a passion to understand the underlying reasons and develop better approaches to help organizations work more effectively. He looks at the world with a socio-technical lens to explore the complex interactions between people systems and technology systems. The knowledge span includes system engineering, business process analysis, quality and performance management system, organization design and behaviour, technology induced change, human psychology and motivation. The application domains include aerospace, engineering, manufacturing, business services, IT, education, health, local government.



Stephen King is currently a part-time senior Lecturer in Advanced Analytics having recently retired from Rolls-Royce (April 2020) where he was an Engineering Associate Fellow and EHM Specialist working within the Rolls-Royce Digital organisation. During his 41-year career at Rolls-Royce he held positions within the Measurement

Engineering group, Electronics and Measurement Techniques department, Strategic Research Centre, Business Process Improvement Centre, Controls Engineering and System Design Engineering. Prior to this he worked for Electronic Flow Meters where he was responsible for the test and commissioning of flow measurement systems in the oil and gas industry. His main interests are in the use of data mining and advanced analytical techniques for asset health monitoring applications. He holds a degree in Mathematics and Computer Science and a PhD in the application of expert systems for vibration analysis. In addition to being a Chartered Engineer, he is a Fellow member of both the Institution of Engineering and Technology and the Institute of Mathematics and its Applications.

An Approach to Condition Monitoring of BLDC Motors with Experimentally Validated Simulation Data

Max Weigert¹

¹*Institute of Flight Systems and Automatic Control, Technical University of Darmstadt, Otto-Berndt-Str. 2, 64287 Darmstadt, Germany*

weigert@fsr.tu-darmstadt.de

ABSTRACT

Due to their compact design and low number of wear parts, Brushless Direct Current (BLDC) motors are ideally suited for use in unmanned aerial vehicles (UAVs). In view of the growing areas of application and the increasing complexity of unmanned flight missions, the need for suitable safety mechanisms for the operation of technical components, such as BLDC motors, in unmanned aircraft drive trains is also increasing. The integration of redundant components analogous to manned aviation is often not possible for smaller unmanned aerial vehicles for weight reasons. Therefore, online-capable dynamic diagnosis and prognosis methods for monitoring safety-critical components of unmanned aircraft are subject of ongoing research.

One major challenge in the development of data based condition monitoring approaches for safety critical components is the availability of operational data of degraded components. This often leads to an unbalanced database without sufficient information on components' degradation behavior.

In the presented work, this problem is approached by combining bench testing and simulation models. On a test rig, common degradation effects are recreated by targeted manipulation. This allows for a safe and expressive data acquisition of the components' behavior. In order to reduce the material and time required to build up a sufficient database for condition monitoring with experimental data, the observable effects are replicated in a simulation. This provides the opportunity to create a large database with slight variations in simulation parameters and incorporated noise in the simulation.

The BLDC motor manipulation on the test rig includes mechanical, electrical and magnetic manipulation. The effects of the manipulation are analyzed and their representation by parameters in the corresponding simulation

is derived. The model is built in MATLAB Simulink and replicates both the electrical and physical behavior of the motor, as well as its commutation behavior.

The established simulation data shall be used as a balanced dataset on which condition monitoring algorithms can be trained. This will allow for the comparison of various data based condition monitoring methods in the future. A remaining challenge lies in the time behavior of the analyzed degradation, which has not yet been explored in depth. The proposed approach might also be applied to further unmanned aerial vehicle components, such as servo motors.

1. INTRODUCTION

A meaningful assessment of the flight risk of UAVs is an important basis towards their increased use and enhanced automatization. The influence of different UAV components on its safety has therefore already been thoroughly discussed in the literature. Shafiee, Zhou, Mei, Dinmohammadi, Karama and Flynn (2021) conducted a fault tree and a failure mode and effects analysis (FMEA) for UAV failures during inspection of offshore wind turbines. In their FMEA, the risk of motor bearing failures was assigned the second highest priority after the risk of lack of battery power, which is not directly connected to the components' health. A similar FMEA was performed by Wang, Ng, Elhadidi, Ang and Moon (2019), who also studied the structure of the UAV with a finite element simulation. They as well assigned the second highest risk to the motor and electronic speed controllers (ESCs) after the battery. Osborne, Lantair, Shafiq, Zhao, Robu, Flynn and Perry (2019) conducted a survey with commercial UAV operators and summarized their assessment of UAV component safety. Their participants rated BLDC motor failures as relatively unlikely but attributed better warning of electrical and mechanical failures as the second highest priority for increasing UAV safety.

Analyzing possible failure mechanisms of BLDC motors is mostly done on corresponding test rigs. At the Institute for Flight Systems and Automatic control, this has already been done by Haus, Mikat, Nowara, Kandukuri, Klingauf, and

Max Weigert. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Buderath (2013) and by Wolfram, Vogel, and Stauder (2018). Haus et al. (2013) conducted a motor current signature analysis for asynchronous motors. They induced motor abnormalities by grinding the inner ring of the bearing with an angle grinder, short-circuiting stator windings together, and creating imbalances in the rotor. Wolfram et al. (2018) built up the test rig used in this work. They aimed for condition monitoring of multicopter drive trains by examining the input and output power of their components. Faults were generated by notches on the propeller and applying diamond paste into the bearing.

An overview of faults in BLDC motors is given by Kudelina, Asad, Vaimann, Rassolkin, Kallaste and Lukichev (2020). They differentiate mechanical failures, most often associated with the bearing, electrical failures and permanent magnet failures. Experimental replication of such failures has also been approached by Shifat and Hur (2020). They introduced short circuits of the BLDC coils, both in the same and in neighboring phases. Yang, Habibullah, Zhang, Xu, Lim and Nadarajan (2016) studied accelerated thermal aging of electrical motors. Repeated heating cycles of motors in an oven could clearly be associated with a health indicator based on their constructed features. Siddiolo and Buderath (2018) developed a prognostic framework for the Remaining Useful Lifetime (RUL) of aeronautic fan ball bearings. For their Run-to-Failure tests, they introduced diamond-powder into the bearing.

The challenge of data-based condition monitoring based on experimental motor studies is the small database generated. To overcome this problem, this work aims at building up a simulation model of a BLDC motor and recreating degradation mechanisms in the simulation. This approach is for example also proposed in (Wolfram et al. 2018) and (Siddiolo and Buderath 2018).

The Simulation of BLDC motor behavior is based on its mathematical description. The underlying physical principles are for example discussed in (Hanselman 1994). A detailed derivation of the voltage profile for pulse width modulation commutation is given in (Baldursson 2005). Zhang, Liu, Peng and Liu (2020) used a mathematical model of a BLDC motor, to compare its healthy state with one involving an increased inverter resistance in one phase. Gupta, Jayaraman and Reddy (2021) as well simulated the BLDC motor behavior with increased resistance values and analyzed its effects on the motor behavior.

2. OPERATION OF THE BLDC MOTORS

This work exemplarily focuses on the components of the SciHunter UAV, which was built as a reference system at the institute. The UAV makes use of two different types of BLDC motors, four Multiplex ROXXY C42 motors are used for lift generation, one Multiplex ROXXY C35 motor is used for thrust generation. Both motor variations are designed as brushless outrunners with 14 magnetic poles on the rotor and

12 coils on the stator. They differ in their dimensions and performance according to their different operating requirements.

The thrust generating Multiplex ROXXY C35 is chosen as a starting point for the motor analysis. It is expected to be operated in the majority of the flight time, as the fixed wing flight with generated thrust is faster and more energy efficient compared to the hover flight. The behavior of the motor is analyzed for a constant thrust generation in uniform flight. This allows for a reduced complexity and good generalization of the data analysis. During flight, the motor is expected to be operated mainly in unaccelerated flight phases at the equilibrium. Therefore, those flight phases are providing both the largest amount of data for a data based condition monitoring and the most frequent opportunity to collect and analyze current flight data. The smaller variation in the sensor data due to the uniform operation of the motor in unaccelerated flight is an additional advantage that simplifies the interpretation of the created database. For a typical flight speed of 60 km/h picked out of the possible fixed wing flight speed range from 50 km/h to 72 km/h of the UAV, the motor has to provide 5.17 N of thrust based on the known drag coefficient of the UAV.

The picked operating point is approximated by having the motor rotate an airscrew generating the desired thrust both in the experimental setup and the created simulation model. The airspeed during flight is neglected for the initial data collection, as it is not expected to significantly affect the effects of degradation on the motor. Both the test rig and the simulation provide the opportunity to integrate operating conditions of higher complexity in the future by airspeed or varying motor speed.

2.1. Experimental Setup

The experimental setup utilizes the existing test rig of the institute. The test rig is visualized in Figure 1. The motor is operated by the Electronic Speed Controller (ESC) FrSky Neuron 60, which is also deployed in the SciHunter. The ESC can be controlled from LabView, where the test procedure is coordinated and the measured data is collected. It is connected to a switched-mode power supply with a DC voltage of 14.8 V, equivalent to the battery voltage of the UAV. For the uniform operation profile, the motor is equipped with its corresponding airscrew and a duty cycle of 48 % is set for the ESC, which results in a rotational motor speed of approximately 3660 rotations per minute (rpm) and the desired thrust of 5.17 N generated by the airscrew.

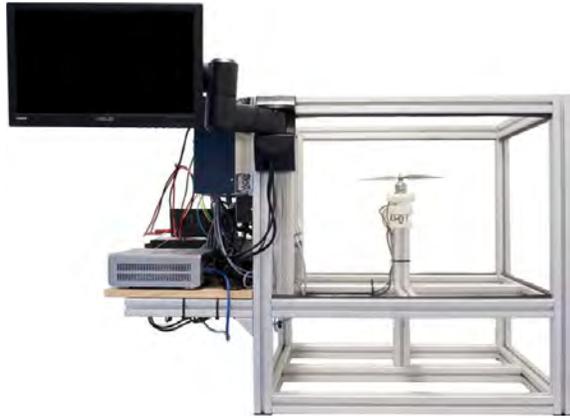


Figure 1. Test rig setup by (Wolfram et al. 2018).

The motor is mounted on top of a force and a torque sensor and a thermocouple is inserted in the stator next to its windings. The rotational speed of the motor is measured by an optical infrared measurement. A measuring board equipped with current and voltage sensors is used to observe the supply current of the ESC and the phase currents from the ESC to the motor. In future measurement series, the telemetry data of the ESC might also be collected and used as a reference for voltage, current and speed measurements.

2.2. Mathematical Modelling

To describe the motor behavior with physical equations, the abstracted circuit diagram in Figure 2 is utilized. It represents the ESC switches used for commutation and the coils of the three different phases, which are connected in a star configuration. Each depicted coil represents 4 stator coils, which are connected in series in the described motor. Compared to the simplest possible coil configuration with one coil per phase and two magnet poles, its back electromotive force (EMF) with 14 magnetic poles is periodic for one seventh of the motor rotation and it is assumed to be ideally trapezoidal shaped. Interactions between the coils are neglected in the presented modelling.

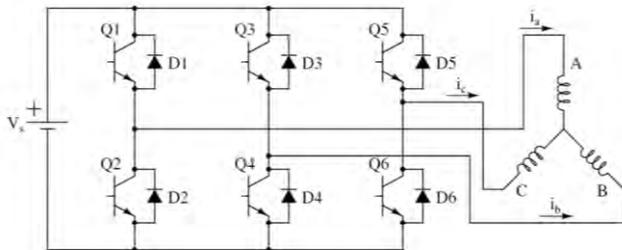


Figure 2. BLDC drive scheme (Baldursson 2005).

The switching behavior of the ESC is based on the current position of the motor. For every seventh of a motor rotation,

6 commutation configurations are gone through. Each of them features one coil with positive, one with negative and one with no applied voltage. The 6 configurations can be summarized by the two variations of the circuit diagram shown in Figure 3, which can both be applied for the three current combinations i_a & i_b , i_b & i_c and i_c & i_a as i_1 & i_2 . Additionally shown in Figure 4 are the two variations of the circuit diagram derived with only one closed switch for each current combination. Those configurations are assumed to be present when the pulse width modulation signal is 0 and the voltage supply shall be temporarily withdrawn. The ratio of active and inactive pulse width modulation signal is specified by the duty cycle. The switches and diodes of the ESC are assumed as ideal and their physical behavior is not accounted for. This allows for the derivation of the voltage difference between each of the phases, depending on the motor position. For a detailed statement of all derivable voltage equations, refer to (Baldursson 2005).

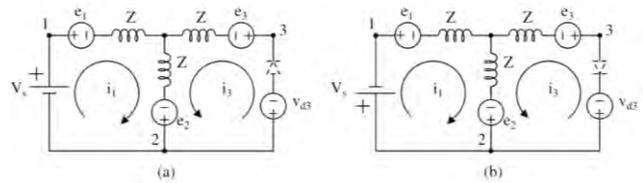


Figure 3. Commutation variations with voltage supply (Baldursson 2005).

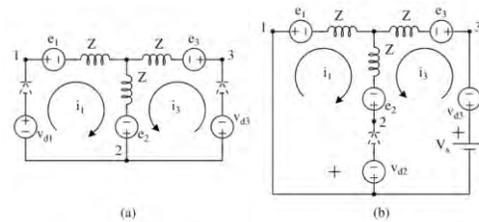


Figure 4. Commutation variants without voltage supply (Baldursson 2005).

The currents in the three phases can then be derived based on the mesh equations between the phases a & b and b & c. In accordance with the star connection, the current of the third phase can be calculated as the negative sum of the other two phases. Therefore, only two currents are calculated with differential equations and only two voltage differences supply sufficient information. Those current derivatives are given in Eq. 1 and Eq. 2 with the voltage differences v_{ab} and v_{bc} already denoting the voltage less the back EMF differences from the respective phases.

$$\begin{aligned} & \frac{d}{dt} * i_a \\ & = ((L_b + L_c) * ve_{ab} + L_b * ve_{bc} \\ & - (L_b * R_a + L_b * R_c + L_c * R_a) \cdot i_a \\ & + (-L_b * R_c + L_c * R_b) * i_b) \\ & * \frac{1}{L_a \cdot L_b + L_a \cdot L_c + L_b \cdot L_c} \end{aligned} \quad \text{Eq. 1}$$

$$\begin{aligned} & \frac{d}{dt} * i_b \\ & = (-L_c * ve_{ab} + L_a * ve_{bc} \\ & + (-L_a * R_c + L_c * R_a) * i_a \\ & - (L_a * R_b + L_a * R_c + L_c * R_b) * i_b) \\ & * \frac{1}{L_a * L_b + L_a * L_c + L_b * L_c} \end{aligned} \quad \text{Eq. 2}$$

The electric moment is calculated based on the obtained phase currents according to Eq. 3. It is in equilibrium with the product of inertia and acceleration, the airscrew torque and the frictional torque.

$$\begin{aligned} T_e & = \frac{e_a * i_a + e_b * i_b + e_c * i_c}{\dot{\Theta}_m} \\ & = \frac{k_e}{2} * (f_a * i_a + f_b * i_b + f_c * i_c) \end{aligned} \quad \text{Eq. 3}$$

The generator constant k_e and the resistance are both approximated as linearly temperature dependent, the generator constant with a negative and the resistance with a positive proportionality factor. The motor temperature difference to the ambient temperature is described in a simplified way by the product of the power loss and the thermal resistance between windings, housing and environment. In the simulation, the time behavior of the motor heating is modeled with a PT_1 transmission behavior. To avoid simulating long warm-up processes, a smaller time constant for the motor startup can be enabled.

3. REPLICATION OF DEGRADATION EFFECTS

To build up an expressive database for the development of data based condition monitoring and prognosis methods, this work aims at replicating degradation effects on the considered component. For this purpose, the regarded motor was manipulated on a test rig and a simulation model of the engine accounting for different degradation effects was built up.

3.1. Test rig

On the test rig, mechanical, electrical and magnetic manipulation of the motor was pursued. To achieve distinctive, traceable, and timely results, active adjustments were made to the motor. Long-term run to failure tests without active advancement of specific degradation

mechanisms might be supplementarily carried out in the future.

3.1.1. Mechanical Manipulation of the Motor

The motor bearings were identified as primary source of mechanical failure. To intensify the bearing degradation by increased friction and abrasion, diamond powder was inserted in the bearing. A run to failure test with the manipulated motor was carried out. The measurement was characterized in particular by a sudden failure of the motor. Cause of the fault was a broken rolling element cage, as depicted in Figure 5. The measurement was conducted with a periodical measurement interval, as a constant degradation trend was expected. Unfortunately, the failure occurred in between two measurement periods. For further analysis, a more sophisticated measurement method is planned, in which measurement data shall first be stored temporarily and then be largely deleted again if there are no significant changes in the system behavior.



Figure 5. Motor with broken bearing cage.

3.1.2. Electrical Manipulation of the Motor

To manipulate the electrical behavior of the motor, two methods have been utilized. A winding short circuit was introduced by a solder drop on a motor coil and the phase resistance was increased by connecting a series resistor.

With the solder drop, covering approximately 20 % of one coil, neighboring windings on multiple winding levels have been short circuited. An image of the manipulated stator is given in Figure 6. In particular, the behavior in the event of local damage to the insulation and increased heat build-up should be simulated, which might particularly affect neighboring windings on several winding levels. The effects of the solder drop are clearly visible on the test bench and the achievable motor speed at the specified duty cycle of 48 % drops from 3660 rpm to 3315 rpm.



Figure 6. Manipulated stator with solder drop.

By introducing various series resistors, degradation effects due to coil material aging or temperature raises with declining efficiency shall be addressed. The effects of the series resistors are clearly visible, leading to an increasing drop in motor speed with increasing series resistance for a fixed duty cycle. The series resistor was connected with one phase of the motor, which lead to a decreased current in that phase, while the currents in the remaining phases slightly increased with decreasing motor speed.

3.1.3. Magnetic Manipulation of the Motor

The magnetic field of the rotor was manipulated by heating the rotor at 220 °C for one hour. The magnetic flux density of the rotor was measured before and after the heating process with a 3-axis magnetic field sensor and approximately halved in size. The effects of the reduced magnetic field resulting in higher currents and a higher resulting motor speed at the same duty cycle are also clearly visible.

3.2. Simulation

A Simulation Model of the motor has been built up in MATLAB Simulink. It relies on the mathematical description of the motor introduced in chapter 2.2. Beside its replication of the nominal motor behavior, 10 kinds of degradation mechanisms can be recreated in the model. Up to now, these mechanisms have only been individually activated for simulations, but they can also be enabled in combination with each other for an even broader database.

To ensure a high amount of unique simulation outputs, the coil inductivities and resistances can be slightly varied for each simulation. Additionally, the ambient temperature can have noise applied to it. This affects the airscrew torque as well as the resistances and generator constant. Therefore, it replicates the dependency of the motor of its environment, which might introduce variations in its behavior on the test rig.

3.2.1. Mechanical Degradation Recreation

To model mechanical degradation in the simulation, four degradation aspects are considered. Those involve increased bearing friction, periodic bearing loads, static eccentricity effects and dynamic eccentricity.

The increasing bearing friction with time can directly be implemented. It depicts the results of contamination in the bearing, loss or exhaustion of the lubricant and deteriorated rolling characteristics due to abrasion. In reference to bearing lifetime estimations by bearing manufacturers, a growth of the friction coefficient is implemented as a function of normalized torque, normalized rotational speed and number of revolutions. With the aim of imitating a realistic bearing lifetime, changes in the defining degradation parameters as the bearing friction coefficient are nearly undetectable within the simulated timespans. For a condition monitoring, the simulation of distinguished degradation stages already offers valuable information, but the time behavior of the degradation effects remains of high interest for a holistic PHM system.

Periodic bearing loads resemble local deterioration effects like pitting. They are defined with a lookup table in Simulink and only present at a certain motor angle. The angle of the periodic bearing loads can be randomly varied for multiple simulations to achieve an extensive database.

Static eccentricity effects and dynamic eccentricity effects are expressed by a change in the trapezoidal functions of the back EMFs. For the dynamic eccentricity, the relative position of the stator axis to the rotor axis remains constant, and the back EMFs of the three phases are changed by static summands. They are derived as multiples of the sine of the eccentricity angle summed with the phase angles 0 , $2/3 \pi$ and $4/3 \pi$. For the static eccentricity, the relative position of the stator axis and the rotor axis change during the motor rotation. Therefore, the static eccentricity is calculated by adding the current rotation angle to the phase angles introduced for the dynamic eccentricity. A combination of both effects by adding summands representing both eccentricities at the same time is not yet implemented. Similar to the periodic bearing load, the angle of attack can be randomly varied to widen the simulated database.

3.2.2. Electrical Degradation Recreation

The degradation of the motor coils is represented by three short-circuit and two resistance increase variations. To broaden the simulation results, the coil numbers in the simulation can be switched. That way, it is ensured, that mechanisms affecting specific phases of the simulated motor can be acted out for all three phases.

The resistance increase in the simulation shall recreate the same effects as in the motor manipulation. In the simulation, increasing the resistance of a single coil specifically represents degradation effects due to local influences like

damaged insulation or polluted contacts, which might lead to higher temperatures and faster material aging. In another variation of the degradation mechanism, by increasing the resistance of all coils uniformly, degradation effects without designated points of attack are tackled.

Similar distinctions are made for the simulation of short circuits. Degradation effects without a specified point of attack are depicted as uniform winding failure. These are modeled under the simplifying assumption that all phases have internal short circuits in equal measure, as illustrated in Figure 7. Their short circuit ratio, expressing the number of short circuited windings divided by the overall winding number in each phase, is increased with the time of fault activity. This shall resemble a progressing failure of the insulations of adjacent windings due to increased heat development. The short circuits are assumed to be ideal, not presenting any resistance and reducing the effective resistance and inductivity with the factor $(1-\mu)$. The back EMF is assumed to be independent of the current flow in the short circuited windings. As only a part of the back EMF is passed by the current in each phase, the short circuit ratio is passed on as a factor to the created electric moment.

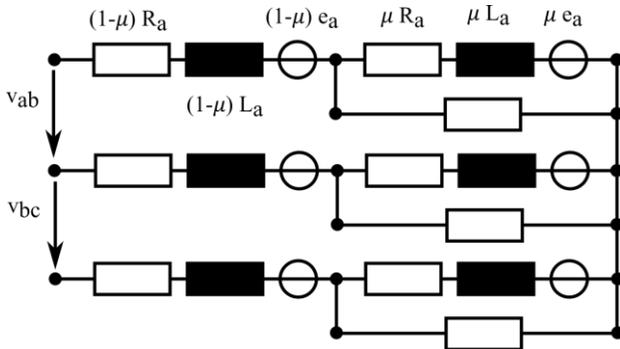


Figure 7. Circuit diagram with uniform short circuits.

For a single phase winding short circuit, the circuit diagram in Figure 8 is derived. The short circuit ratio is assumed to be at a fixed position, where the insulation might be compromised. With time, the resistance of the short circuit is expected to decrease towards zero. A new equation system describing the currents is derived from the circuit diagram, as given in Eq. 4 to Eq.6. It includes the short circuit current i_s . The electric moment is then calculated with respect to the effective currents passing the respective back EMF windings. For the short circuited phase, the product of the short circuit ratio and the short circuit current is subtracted from the phase current, as it resembles the share of back EMF, which is not passed by the short circuit current.

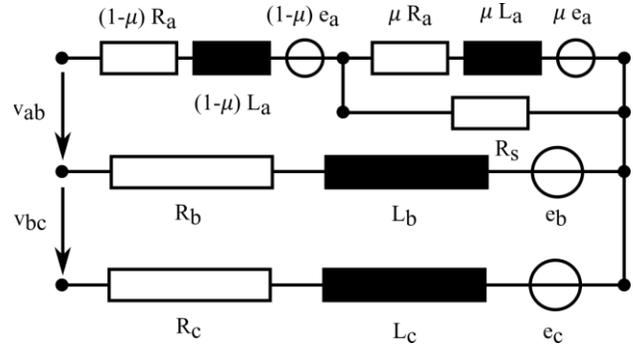


Figure 8. Circuit diagram with one phase short circuited.

$$\begin{aligned} \frac{d}{dt} * i_1 &= ((L_2 + L_3) * ve_{12} \\ &+ L_2 * ve_{23} \\ &+ (-L_2 + L_3) * R_1 * (1 - \mu) - L_2 * R_3) * i_1 \\ &+ (-L_2 * R_3 + L_3 * R_2) * i_2 \\ &+ (-L_2 + L_3) * R_s * i_s \\ &+ (L_2 + L_3) * e_1 * \mu \\ & * \frac{1}{L_1 * (1 - \mu) * (L_2 + L_3) + L_2 * L_3} \end{aligned} \quad \text{Eq. 4}$$

$$\begin{aligned} \frac{d}{dt} * i_2 &= (-L_3 * ve_{12} \\ &+ L_1 * (1 - \mu) * ve_{23} \\ &+ (-L_1 * (1 - \mu) * R_3 + L_3 * R_1 * (1 - \mu)) * i_1 \\ &+ - (L_1 * (1 - \mu) * (R_2 + R_3) + L_3 * R_2) * i_2 \\ &+ L_3 * R_s * i_s \\ &+ - L_3 * e_1 * \mu \\ & * \frac{1}{L_1 * (1 - \mu) * (L_2 + L_3) + L_2 * L_3} \end{aligned} \quad \text{Eq. 5}$$

$$\begin{aligned} \frac{d}{dt} * i_s &= ((L_2 + L_3) * ve_{12} \\ &+ L_2 * ve_{23} \\ &+ \left(-L_2 * R_3 + \frac{L_2}{L_1} * L_3 * R_1\right) * i_1 \\ &+ (-L_2 * R_3 + L_3 * R_2) * i_2 \\ &+ \left(-\left(L_2 + L_3\right) + \frac{L_2}{L_1} * L_3\right) * \left(\frac{R_s}{\mu} + R_1\right) \\ &+ (L_2 + L_3) * R_1 * \mu * i_s \\ &+ \left((L_2 + L_3) + \frac{L_2}{L_1} * L_3\right) * e_1 \\ & * \frac{1}{L_1 * (1 - \mu) * (L_2 + L_3) + L_2 * L_3} \end{aligned} \quad \text{Eq. 6}$$

A two phase winding short circuit is represented by the circuit diagram in Figure 9. Similar to the single phase short circuit, a new equation system can be derived, which accounts for the short circuit current as stated in Eq. 7 to Eq. 9. The short circuit ratios of the two phases are set as equal for the current simulations. The effective currents for the electric moment

calculation include the short circuit current in both affected phases factored with the respective short circuit ratio and opposing signs.

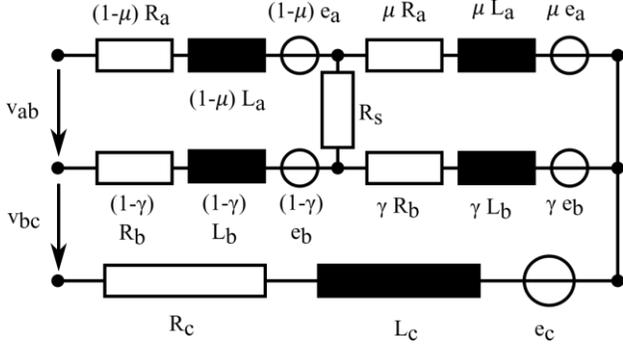


Figure 9. Circuit diagram with two phases short circuited.

$$\begin{aligned}
 & \frac{d}{dt} * i_a \\
 & = ((-L_2 * \gamma)^2 + (L_1 * \mu + L_2 * \gamma) * (L_2 \\
 & + L_3)) * v_{e12} \\
 & + L_2 * (1 - \gamma) * (L_1 * \mu + L_2 * \gamma) * v_{e23} \\
 & + (-L_2 * \gamma * ((R_1 + R_3) * L_2 * (1 - \gamma) + L_3 \\
 & * R_1 * (1 - \mu)) + L_1 * \mu * ((L_2 + L_3) * R_1 * (1 \\
 & - \mu) + L_2 * (1 - \gamma) * R_3)) * i_1 \\
 & + ((-R_3 * (L_1 * \mu + L_2 * \gamma) + L_3 * R_2 * (L_1 \\
 & * \mu / L_2 + \gamma)) * L_2 * (1 - \gamma)) * i_2 \\
 & + (-L_1 * \mu * R_s * (L_2 + L_3) + L_2 * \gamma \\
 & * (L_2 * (1 - \gamma) * R_1 * \mu - L_1 * \mu * R_2 * (1 - \gamma) \\
 & - L_3 * R_s)) * i_s \\
 & + (L_1 * \mu * (L_2 + L_3) + L_2 * \gamma * L_3) * (e_1 * \mu \\
 & - e_2 * \gamma) \\
 & / \left(L_1 * \mu * \left(\frac{(L_2 + L_3) * L_1 * (1 - \mu)}{+ L_3 * L_2 * (1 - \gamma)} \right) + L_2 * \gamma \right. \\
 & \left. * \left(\frac{(L_1 + L_3) * L_2 * (1 - \gamma)}{+ L_3 * L_1 * (1 - \mu)} \right) \right)
 \end{aligned}$$

Eq. 7

$$\begin{aligned}
 & \frac{d}{dt} * i_b \\
 & = ((-L_1 * \mu * L_2 * \gamma - (L_1 * \mu + L_2 * \gamma) * L_3) \\
 & * v_{e12} \\
 & + (L_1 * (1 - \mu) * (L_1 * \mu + L_2 * \gamma)) * v_{e23} \\
 & + ((L_1 * \mu + L_2 * \gamma) \\
 & * (L_3 * R_1 * (1 - \mu) - L_1 * (1 - \mu) * R_3)) * i_1 \\
 & + (-L_1 * (1 - \mu) * (L_1 * \mu * (R_2 + R_3) + L_2 \\
 & * R_3 * \gamma) - R_2 * (1 - \gamma) * ((L_1 + L_3) * L_2 * \gamma \\
 & + L_1 * L_3 * \mu)) * i_2 \\
 & + (L_1 * \mu * (-L_1 * (1 - \mu) * R_2 * \gamma + L_2 * \gamma \\
 & * R_1 * (1 - \mu)) + R_s * ((L_1 * \mu + L_2 * \gamma) * L_3 \\
 & + L_1 * L_2 * \gamma)) * i_s \\
 & + (-L_1 * L_2 * \gamma + (L_1 * \mu + L_2 * \gamma) * L_3) \\
 & * (e_1 * \mu - e_2 * \gamma) \\
 & / \left(L_1 * \mu * \left(\frac{(L_2 + L_3) * L_1 * (1 - \mu)}{+ L_3 * L_2 * (1 - \gamma)} \right) + L_2 * \gamma \right. \\
 & \left. * \left(\frac{(L_1 + L_3) * L_2 * (1 - \gamma)}{+ L_3 * L_1 * (1 - \mu)} \right) \right) \\
 & \frac{d}{dt} * i_s \\
 & = ((L_1 * L_2 * \mu + (L_1 * \mu + L_2 * \gamma) * L_3) \\
 & * v_{e12} \\
 & + (L_1 * \mu * L_2 - L_1 * L_2 * \gamma) * v_{e23} \\
 & + (L_2 * (\gamma - \mu) * (L_1 * R_3 - L_3 * R_1)) * i_1 \\
 & + (L_1 * (\mu - \gamma) * (L_3 * R_2 - L_2 * R_3)) * i_2 \\
 & + (-L_1 * (1 - \mu) * ((L_2 + L_3) * R_1 * \mu + L_3 \\
 & * R_2 * \gamma) + R_s * (L_1 * (L_2 + L_3) + L_2 * L_3) \\
 & + L_2 * (1 - \gamma) * ((L_1 + L_3) * R_2 * \gamma + L_3 * R_1 \\
 & * \mu)) * i_s \\
 & + (L_1 * (L_2 + L_3) + L_2 * L_3) * (e_1 * \mu - e_2 \\
 & * \gamma) \\
 & / \left(L_1 * \mu * \left(\frac{(L_2 + L_3) * L_1 * (1 - \mu)}{+ L_3 * L_2 * (1 - \gamma)} \right) + L_2 * \gamma \right. \\
 & \left. * \left(\frac{(L_1 + L_3) * L_2 * (1 - \gamma)}{+ L_3 * L_1 * (1 - \mu)} \right) \right)
 \end{aligned}$$

Eq. 8

Eq. 9

3.2.3. Magnetic Degradation Recreation

To represent a decrease in the magnetic flux density, which might be caused by too high motor temperatures or mechanical stresses, decreased trapezoidal functions of the back EMFs are applied in the simulation. This is assumed to be an adequate representation of magnetic degradation, as the back EMF can be described as a function of the coil dimensions, the magnetic flux density and the relative position of rotor and stator.

4. OBSERVED BEHAVIOR

In order to investigate the validity of the simulation environment, the observed engine behavior on the test rig is compared with that from the simulation. Figure 10 and Figure 11 depict the phase currents on the test rig and in the simulation under healthy conditions. Both are shown under uniform operating conditions for a small time window, which allows distinguishing the commutation phases visible in the data. The model parameters have not yet been adjusted to match the test rig values, but the same behavior pattern is recognizable in both graphs. On the test rig, the commutation behavior of the ESC cannot be resolved in time though, while its resulting current fluctuations can be clearly seen in the simulation.

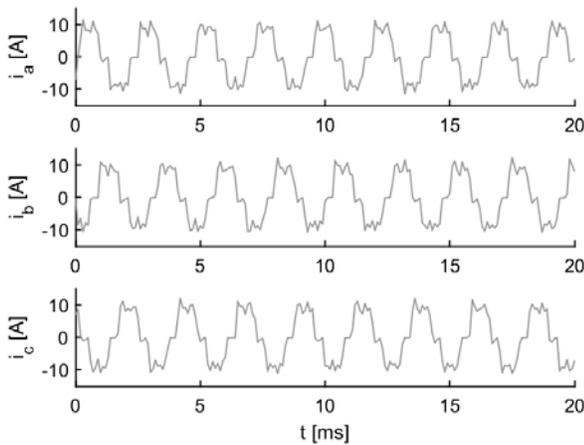


Figure 10. Test rig motor currents in healthy state.

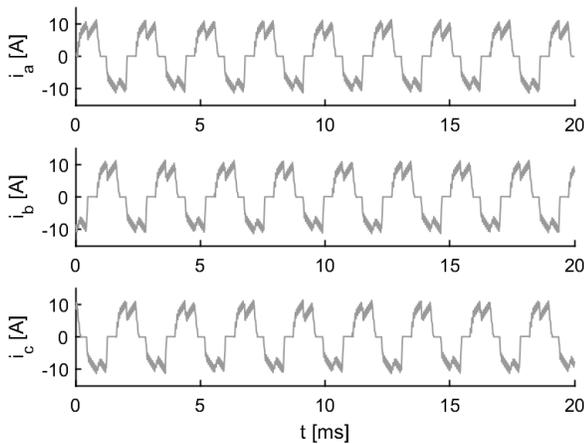


Figure 11. Simulation motor currents in healthy state.

With regard to the represented degradation mechanisms, the electrical failure mode of an increased resistance and the magnetic failure mode are shown as examples. As described in chapter 3.1.1, the behavior of the mechanical failure mode

could not be recorded completely and will therefore be investigated in future by means of further series of measurements. The failure mode behavior on the test rig and in the simulation are illustrated in Figure 12 to Figure 15 by the measured phase currents of a motor operated at the same duty cycle, as in the healthy state.

For both failure mechanisms, the observed behavior in experiment and simulation are well comparable. With the increased resistance in a single phase, a drop in the affected phase currents becomes visible. Simultaneously, a drop in the rotational speed occurs at a constant duty cycle. The decreased magnetic flux density leads to a slight increase in the rotational speed accompanied by an increase in the three phase currents.

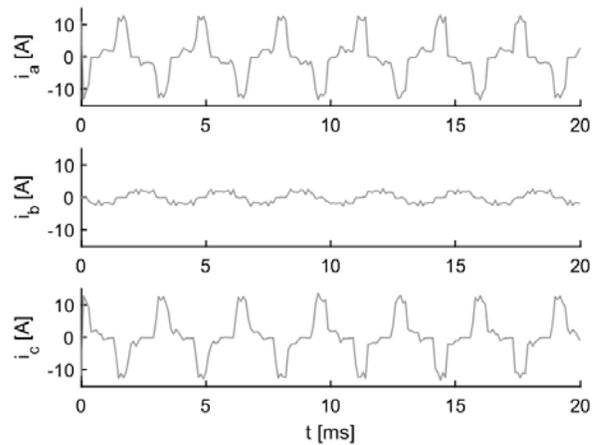


Figure 12. Test rig motor currents with increased resistance.

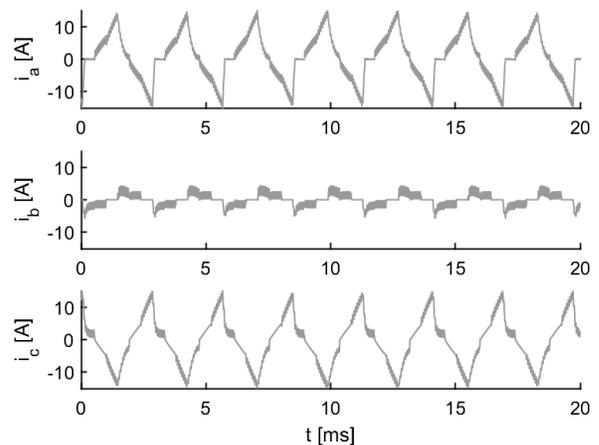


Figure 13. Simulation motor currents with increased resistance.

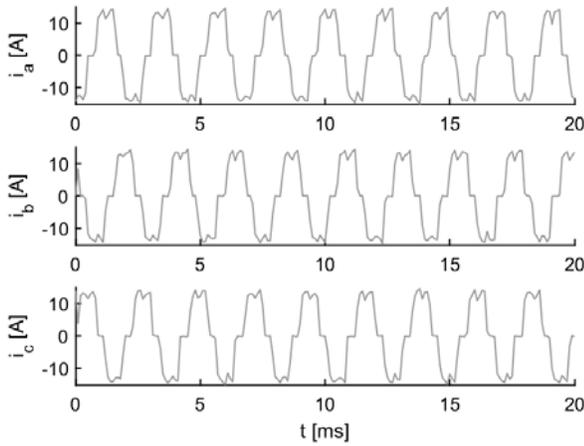


Figure 14. Test rig motor currents with demagnetization.

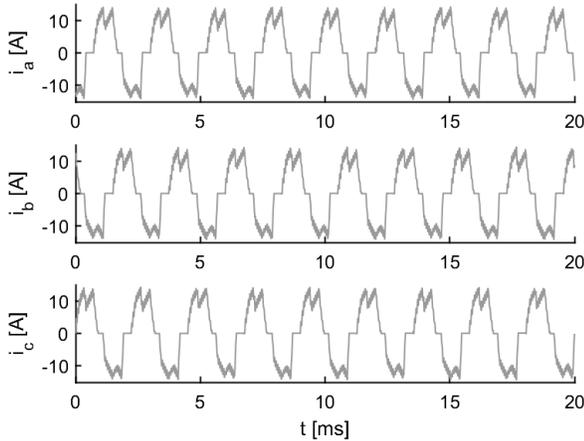


Figure 15. Simulation motor currents with demagnetization.

5. CONCLUSION AND OUTLOOK

In this paper, an approach is discussed to represent failure modes of BLDC motors experimentally and simulatively. This approach offers the potential to build up a large database for the application of data-based condition monitoring for BLDC engines. The presented work represents only an intermediate state. A more precise adaptation of the simulation parameters to reproduce the experimental results should increase their significance in the future. This starting point will then be used to derive meaningful features and to check the distinguishability of the considered fault cases by means of different machine learning methods.

The developed approach may then also enable a similar representation of the servo motor behavior, which are also safety-relevant components of the hybrid SciHunter UAV. The temporal relationship of the discussed degradation

mechanisms also requires further investigation. Longer-term Run-to-Failure tests may be conducted to investigate this in the future.

ACKNOWLEDGEMENT

This project is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), whose support is highly appreciated by the author. Furthermore, the author would like to thank Mr. André Klütsch for his contributions to the test rig trials in the context of his master's thesis and Mr. Lorenz Dingeldein, for the exchange of simulation approaches for BLDC motors.

NOMENCLATURE

e	back EMF
f	trapezoidal function
k_e	generator constant
L	inductivity
R	resistance
t	time
T_e	electric moment
v	voltage
ve	difference of voltage and back-EMF
μ	first phase short circuit ratio
γ	second phase short circuit ratio
θ	motor angle

REFERENCES

- Baldursson, Stefán (2005): BLDC Motor Modelling and Control – A Matlab®/Simulink® Implementation. Chalmers University of Technology, Göteborg.
- Gupta, Annima; Jayaraman, Kalaiselvi; Reddy, Ravula Sugunakar (2021): Performance Analysis and Fault Modelling of High Resistance Contact in Brushless DC Motor Drive. In : IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society. IECON 2021 - 47th Annual Conference of the IEEE Industrial Electronics Society. Toronto, ON, Canada, 13.10.2021 - 16.10.2021: IEEE, pp. 1–6.
- Hanselman, Duane C. (Ed.) (1994): Brushless permanent-magnet motor design. New York: McGraw-Hill, Inc.
- Haus, Steffen; Mikat, Heiko; Nowara, Martin; Kandukuri, Surya; Klingauf, Uwe; Buderath, Matthias (2013): Fault Detection based on MCSA for a 400Hz Asynchronous Motor for Airborne Applications. In : International Journal of Prognostics and Health Management, vol. 4.
- Kudelina, Karolina; Asad, Bilal; Vaimann, Toomas; Rassolkin, Anton; Kallaste, Ants; Lukichev, Dmitri V. (2020): Main Faults and Diagnostic Possibilities of BLDC Motors. In : 2020 27th International Workshop on Electric Drives: MPEI Department of Electric Drives 90th Anniversary (IWED). 2020 27th International Workshop on Electric Drives: MPEI Department of

Uncertainty Informed Anomaly Scores with Deep Learning: Robust Fault Detection with Limited Data

Jannik Zraggen¹, Gianmarco Pizza², and Lilach Goren Huber³

^{1,3} *Zurich University of Applied Sciences, Technikumstrasse 9, Winterthur, 8400 Switzerland*

jannik.zraggen@zhaw.ch

lilach.gorenhuber@zhaw.ch

² *Nispera AG, Hornbachstrasse 50, CH-8008 Zurich, Switzerland*

gianmarco.pizza@nispera.com

ABSTRACT

Quantifying the predictive uncertainty of a model is an important ingredient in data-driven decision making. Uncertainty quantification has been gaining interest especially for deep learning models, which are often hard to justify or explain. Various techniques for deep learning based uncertainty estimates have been developed primarily for image classification and segmentation, but also for regression and forecasting tasks. Uncertainty quantification for anomaly detection tasks is still rather limited for image data and has not yet been demonstrated for machine fault detection in PHM applications.

In this paper we suggest an approach to derive an uncertainty-informed anomaly score for regression models trained with normal data only. The score is derived using a deep ensemble of probabilistic neural networks for uncertainty quantification. Using an example of wind-turbine fault detection, we demonstrate the superiority of the uncertainty-informed anomaly score over the conventional score. The advantage is particularly clear in an "out-of-distribution" scenario, in which the model is trained with limited data which does not represent all *normal* regimes that are observed during model deployment.

1. INTRODUCTION

Assessing the predictive uncertainty of machine learning (ML) and deep learning (DL) algorithms is essential for any decision taken on the basis of such algorithms. Some popular examples for taking decisions under uncertainty include image classification for autonomous-driving (Kraus & Dietmayer, 2019; He, Zhu, Wang, Savvides, & Zhang, 2019; Miller, Day-

oub, Milford, & Sünderhauf, 2019) or for medical purposes (Leibig, Allken, Ayhan, Berens, & Wahl, 2017; Herzog, Murina, Dürr, Wegener, & Sick, 2020) as well as time series forecasting models (Laptev, Yosinski, Li, & Smyl, 2017).

The applications of uncertainty quantification (UQ) to machine learning anomaly detection are still rare, and these focus mostly on anomaly detection in images (Seeböck et al., 2019; Cai, Lu, & Sato, 2020; Sato, Hama, Matsubara, & Uehara, 2019). In time series data, and in particular for machine sensor data, DL based UQ has been primarily used for prognostics models aimed at the estimation of remaining useful life (Biggio, Wieland, Chao, Kastanis, & Fink, 2021). Combining uncertainty estimates in the most fundamental (and application relevant) step of machine fault detection is still missing. As condition-based maintenance often relies on the output of anomaly detection algorithms, uncertainty of such algorithms is necessarily propagated onto uncertainty in maintenance decisions.

In this paper we introduce a method to incorporate the uncertainty quantification of a DL model into an anomaly score. In particular, we suggest to use a regression-based anomaly detection model, in which a model is trained with normal data exclusively and anomalies are detected in the test data based on the deviations (residuals) of the true measurements from the model predictions. Using such a deep regression-based anomaly detection model, the UQ is carried out similarly to a standard regression task, independent of the anomaly detection step. In a subsequent step we derive an anomaly score that combines information about the prediction error together with the prediction uncertainty. We show that the uncertainty-informed anomaly score outperforms the conventional uncertainty agnostic score especially under difficult training conditions, when the training data is not representative for all testing conditions. This scenario is very common for PHM applications, in which machine data is often collected over a

Jannik Zraggen et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

limited period of time prior to commercial deployment (Fink et al., 2020). Despite this, new operating conditions should in general not be detected as anomalies, but as a healthy "out-of-distribution" behaviour. In this sense, exploiting uncertainty information in anomaly detection is more challenging than in classification or forecasting tasks. We show that uncertainty-informed anomaly scores can distinguish between true anomalies and unknown but healthy conditions. An important advantage of the uncertainty-informed score, is that there is no need to use an uncertainty-based filter of the predicted outputs, in order to disqualify or discard the most uncertain predictions, as commonly done in classification or segmentation tasks (Abdar et al., 2021; Schwaiger, Sinhamahapatra, Gansloser, & Roscher, 2020). Instead, each and every prediction obtains an anomaly score and its health condition is assessed given a detection threshold.

There are various approaches for UQ with DL models (Gawlikowski et al., 2021; Abdar et al., 2021). Some methods are based on training an ensemble of networks and using the variance of predictions as a measure for uncertainty (Lakshminarayanan, Pritzel, & Blundell, 2017), other focus on variational inference using MC-Dropout as an estimate for model uncertainty (Gal & Ghahramani, 2016). In this paper we choose to focus on deep ensemble methods. However, since our neural network includes dropout layers for regularization, we in fact combine MC-dropout with ensembling.

In the first part of the paper we focus on the selection of a useful UQ method for our problem. A useful uncertainty measure is on one hand sharp enough to be informative and on the other hand does not suffer from over-confidence, i.e is well calibrated (Kuleshov, Fenner, & Ermon, 2018). The calibration of an uncertainty estimate can be quantified (Kuleshov et al., 2018; Levi, Gispan, Giladi, & Fetaya, 2019; Tran et al., 2020), and the model can be recalibrated in various ways if needed (Kuleshov et al., 2018; Levi et al., 2019). In order to select a properly calibrated model, we contrast the performance of two models: an ensemble of CNN models trained with a Mean Squared Error (MSE) loss is compared with an ensemble of probabilistic CNN models trained with a Negative Log Likelihood (NLL) loss. In the latter case the output of the network includes a mean and a variance of the conditional distribution function (Dürr, Sick, & Murina, 2020). Using an example aimed at wind-turbine fault detection, we demonstrate that the NLL-based ensemble provides a well calibrated uncertainty estimate, as opposed to the MSE-based ensemble. This conclusion is similar to the one in (Lakshminarayanan et al., 2017), however we quantify it here in several different ways.

The second part of the paper is dedicated to using the uncertainty informed regression model for an anomaly detection task. After selecting a reliable uncertainty measure we use it for the derivation of an uncertainty-informed anomaly

score. We show that such a score can improve the fault detection performance compared to standard uncertainty-agnostic scores, particularly when the healthy training data is limited and does not cover all possible (healthy) operational conditions observed during testing. This approach to anomaly detection is the main contribution of the paper and has a potential impact beyond the specific application to wind turbine condition-based maintenance that we provide here as an example.

2. INTRODUCTION TO THE USE-CASE: WIND TURBINE FAULT DETECTION

We demonstrate the usefulness of the uncertainty-informed anomaly score on 4 years of real operational data from the Supervisory Control and Data Acquisition (SCADA) system of a wind turbine. The data contains time series with 10-minute averaged values of environmental and operational variables. The fault detection task is aimed at detecting anomalous patterns in the temperature measurements of various turbine components (Tautz-Weinert & Watson, 2016), focusing primarily on heating rather than cooling effects (one-sided deviations of the temperature). This is achieved by using the component temperature at time t as a target variable y_t in a regression setup with the wind speed, ambient temperature, output power and rotational speed as model inputs. Training the model with data from healthy conditions exclusively, we expect large regression residuals (prediction errors) to be correlated with anomalous behavior. In a previous publication we showed the advantage of using a Convolutional Neural Network (CNN) for this task, and specified our selected architecture (Ulmer, Jarlskog, Pizza, Manninen, & Goren Huber, 2020). Here we repeat only details that are necessary for the performance evaluation of the uncertainty-informed anomaly score.

In the example shown throughout the paper we select the gearbox bearing temperature of the wind turbine as the target variable y_t , in which anomalies are to be detected. The predicting variables are the four mentioned above. The regression CNNs are aimed at providing an uncertainty quantification along with every prediction \hat{y}_t of the bearing temperature at time step t .

A standard approach to anomaly detection based on normal state modeling is to assign anomaly scores to each prediction and set a threshold, above which a prediction is considered anomalous. The conventional anomaly scores are based on the magnitude of the prediction residuals. For example, the anomaly score of a test point at time t can be related to the Cumulative Distribution Function (CDF) of the training residuals, evaluated at the residual $r_t = y_t - \hat{y}_t$ of point t (Clifton et al., 2008):

$$S_t^{(0)} = F(r_t; \mu^{(tr)}, \sigma^{(tr)}) \quad (1)$$

where the mean $\mu^{(tr)}$ and standard deviation $\sigma^{(tr)}$ are esti-

mated from the distribution of the residuals of the entire training data set. The Gaussian CDF is defined as

$$F(y; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^y e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx. \quad (2)$$

In this way, a test point whose residual strongly exceeds the typical training residuals will be detected as an anomaly based on its dissimilarity with the training data. Naturally, this approach is bound to perform less well in case the test data is not well represented in the training set. This applies also when the test data is healthy, i.e with no anomalies. The result in this case may be frequent false positives, leading to unnecessary alarms. In this context it is important to distinguish between such "out of distribution (OoD)" normal data in contrast to true anomalies (e.g machine faults). The main purpose of the uncertainty-informed anomaly score we introduce here is to be able to distinguish between the two, thereby detecting the true anomalies and minimizing the false alarms due to "normal" OoD data.

3. USEFUL UNCERTAINTY QUANTIFICATION

The anomaly detection task essentially decomposes into two sequential steps: (i) a supervised prediction model trained with normal data only (ii) a clustering task of the mixed (normal and abnormal) data, based on anomaly scores assigned to each prediction.

In decision making problems it is beneficial to quantify the uncertainty inherent to the prediction step (i). There are two main sources for uncertainty; aleatoric and epistemic uncertainty (Dürr et al., 2020). Aleatoric uncertainty is also known as data uncertainty and refers to the inherent ambiguity present in the data. Epistemic uncertainty, on the other hand, is known as model uncertainty and is caused by a lack of knowledge of our model.

By including an uncertainty quantification, the prediction model provides not only a single predicted value \hat{y}_t , but an effective predictive distribution, $f(\tilde{\mu}_t, \tilde{\sigma}_t)$, where $\tilde{\mu}_t$ provides an estimate for the predicted value and $\tilde{\sigma}_t$ and estimate for the prediction uncertainty at step t . Since the prediction model is trained with normal data, we expect the predictive distribution of a regression model not to depend on the true value y_t at test time, that is to be independent of whether the ground truth is normal or abnormal. This observation allows us to use for step (i) standard frameworks for UQ commonly used for regression models, ignoring at this point the fact that our ultimate goal is to use this UQ for the anomaly detection task.

In the following we compare different UQ methods in order to select the most useful one. A useful UQ is capable of providing reliable uncertainty estimates for the model predicted output, which is on one hand sharp enough and on the other hand does not suffer from over-confidence (Kuleshov et al., 2018). Selecting a reliable (calibrated) uncertainty quantifi-

cation is relevant for any prediction model, independent of the anomaly detection task following the prediction step.

Similarly to other regression tasks, the purpose here is to identify the most reliable uncertainty measure amongst possible candidates. In this paper we focus on ensemble-based methods for uncertainty estimates. As ensemble members we select CNNs that have been proven to perform well on the anomaly detection task for wind turbines in our previous work (Ulmer et al., 2020). These CNNs already include dropout layers for regularization, which we retain also here. This implies that our UQ is based on deep ensembles with dropout, which is turned on also at prediction time. We thus generate an ensemble of different dropout configurations, where each member of the ensemble is initialized and trained individually. We compare the uncertainty quantifications of two types of CNN ensembles:

MSE ensemble. We train an ensemble of $M = 30$ CNNs by minimizing the prediction MSE. We denote the weights of the m^{th} trained model with θ_m and the predicted value at step t with \hat{y}_{t,θ_m} . For every time step we use the ensemble mean as the prediction and the variance over the ensemble as the uncertainty measure:

$$\begin{aligned} \tilde{\mu}_t &= \frac{1}{M} \sum_{m=1}^M \hat{y}_{t,\theta_m} \\ \tilde{\sigma}_t^2 &= \frac{1}{M-1} \sum_{m=1}^M (\hat{y}_{t,\theta_m} - \tilde{\mu}_t)^2 \end{aligned} \quad (3)$$

NLL ensemble. We train an ensemble of $M = 30$ CNNs by minimizing the prediction NLL. Each member m of the ensemble outputs a predictive distribution $N(\hat{\mu}_{t,\theta_m}, \hat{\sigma}_{t,\theta_m})$. In order to combine the predictive distributions of the NLL-ensemble members we sample a value \hat{s}_{t,θ_m} from the predicted distribution for each step t and each ensemble member m . The estimated mean and uncertainty of the prediction are then defined as:

$$\begin{aligned} \tilde{\mu}_t &= \frac{1}{M} \sum_{m=1}^M \hat{s}_{t,\theta_m} \\ \tilde{\sigma}_t^2 &= \frac{1}{M-1} \sum_{m=1}^M (\hat{s}_{t,\theta_m} - \tilde{\mu}_t)^2 \end{aligned} \quad (4)$$

Note that the variance $\tilde{\sigma}_t^2$ of the sampled values is necessarily larger than the variance of the mean predictions of the same ensemble.

The MSE-ensemble uses the empirical variance of non probabilistic predictions of the CNNs as a measure of uncertainty. This is done differently in the NLL-ensemble. Here each member of the NLL-ensemble models the inherent ambiguity

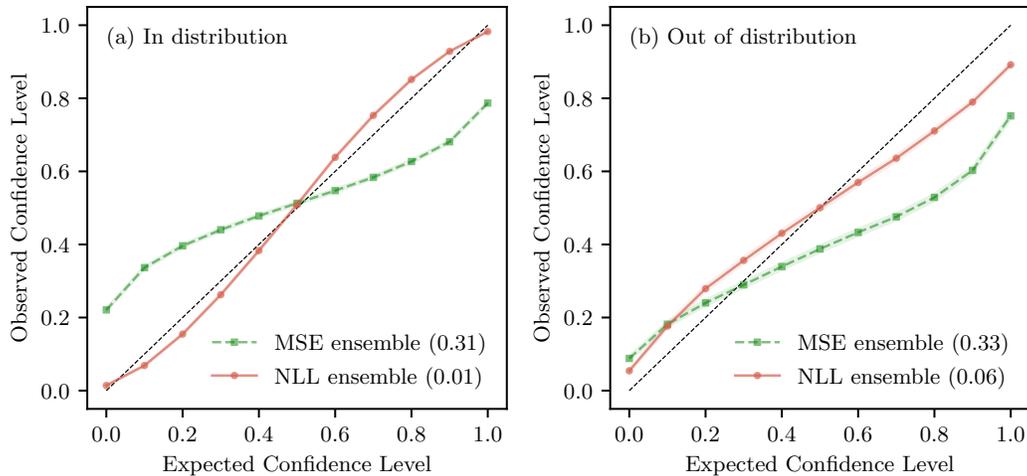


Figure 1. Uncertainty calibration curves. Two uncertainty quantification methods, MSE-ensemble and NLL-ensemble, are compared on two test sets for the prediction of the gearbox bearing temperature: (a) in distribution: a full year healthy test set with both models trained with a full year of healthy data (b) out of distribution: a winter healthy test set with both models trained using healthy summer data. The numbers in brackets are the calculated calibration errors ϵ_{cal} . In both cases the NLL-ensemble model is better calibrated than the MSE-ensemble, and achieves a very low calibration error in distribution.

present in the data (aleatoric uncertainty) and the ensembling over these probabilistic predictions approximates the model uncertainty (epistemic uncertainty). We choose to contrast these two approaches for UQ, despite the inherent difference between them, as the former has been widely used and even claimed in the past to outperform other UQ methods for various applications (Lakshminarayanan et al., 2017).

3.1. Uncertainty Calibration Curves

To assess the calibration level of an uncertainty quantification method we use calibration curves (Kuleshov et al., 2018; Tran et al., 2020). A calibration curve compares the true fraction of points in a given confidence interval with the predicted fraction of points in that interval. Following (Kuleshov et al., 2018), for a given test data set $t = 1 \dots T$ we choose n confidence levels $0 \leq p_1 < p_2 < \dots < p_n \leq 1$ and calculate for each threshold p_j the empirical fraction of true values below it,

$$\hat{p}_j = \frac{\sum_{t=1}^T \mathbb{I}\{y_t \leq F_t^{-1}(p)\}}{T}. \quad (5)$$

The calibration curve is composed of the pairs $\{(p_j, \hat{p}_j)\}_{j=1}^n$. To further quantify the comparison we calculate the calibration error (Kuleshov et al., 2018)

$$\epsilon_{cal} = \sum_{j=1}^n (p_j - \hat{p}_j)^2. \quad (6)$$

Figure 1 shows the calibration curves (including Wilson confidence intervals) for the gearbox bearing temperature predictions of a wind turbine during periods of healthy condi-

tion (no faults). The calibration levels of the two UQ methods, MSE- and NLL-ensemble, are compared, with the calibration errors ϵ_{cal} given in brackets in the legend. In panel (a) the models were trained with data from a full year and the curves were calculated for a time period of one full year. The results demonstrate the clear advantage of UQ using the NLL-ensemble approach which seems to be very well calibrated, with a calibration error of 0.01 (compared to 0.31 for the MSE-ensemble). Note that the shape of the MSE-ensemble curve indicates that this quantification tends to be over-confident, for which the true y_t often falls outside the expected confidence band. The NLL-ensemble method, on the other hand, tends towards a slight under-confidence.

Figure 1(b) repeats the comparison in an Out-of-Distribution (OoD) scenario. It is important to clarify the meaning of OoD in the context of our fault detection task. A common scenario in fault detection applications is that not all healthy (normal) operating conditions are observed during training. As a result, some of these conditions may be detected as anomalies during deployment, only because they are out of the training distribution. Here we use the term OoD to describe these normal operating conditions that *have not been observed during training but should not be detected as anomalies*. In order to emulate such a scenario, we intentionally remove part of the operational conditions from our training set, and introduce these conditions only at testing. Thus, in Figure 1(b) the models are trained with only 3 months of summer data and the calibration curves are calculated on 3 months in winter, where both periods are known to be normal with no anomalies. Since the test data here is clearly OoD (this will be

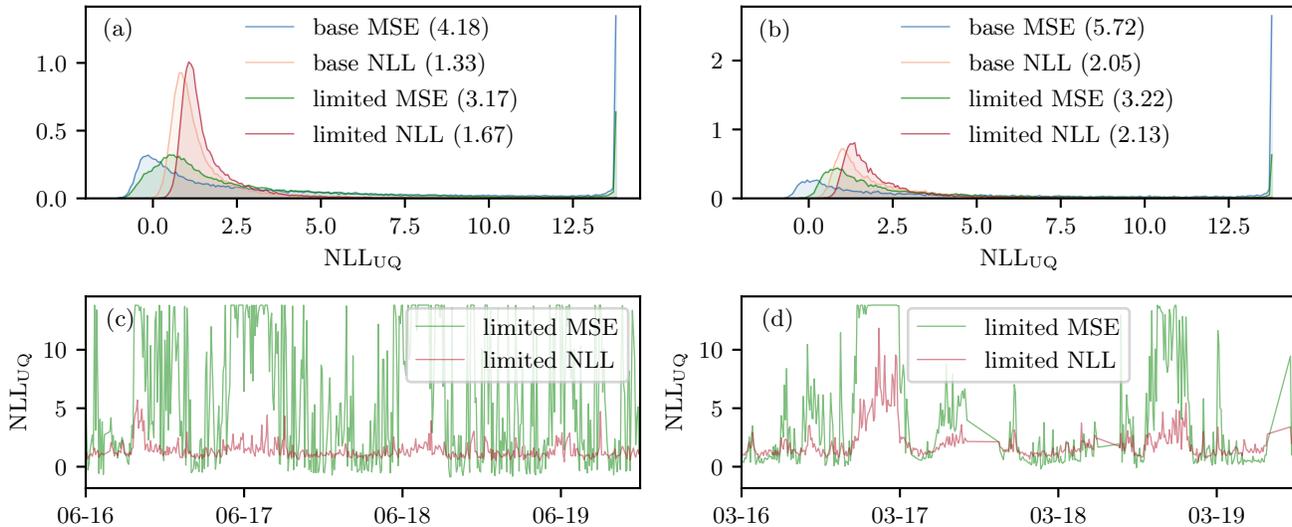


Figure 2. Comparison of the calculated Negative Log Likelihood NLL_{UQ} of the uncertainty quantification for the gearbox bearing temperature prediction. (a) Distributions of the NLL_{UQ} score during one year testing for 4 different models: base MSE is the CNN ensemble with the MSE loss trained with a full year data; base NLL uses the NLL loss with 1 year training data; limited MSE uses the MSE loss trained with 3 months of summer data; limited NLL uses the NLL loss with 3 months summer data for training. The number in brackets is the mean NLL_{UQ} over the entire test period. (b) The same as (a) just for 3 months of test data. The test data is healthy winter data, that is out of the training distribution (OoD) for "limited" training case. (c) An example of the NLL_{UQ} vs. time during summer (in distribution). (d) An example of the NLL_{UQ} vs. time during winter (OoD). The NLL-ensemble generally obtains lower values than the MSE-ensemble when both are trained with limited summer data.

demonstrated again below through increased OoD residuals in Figure 4), the UQ of both models is less well calibrated and both suffer from over-confidence. However, the NLL-ensemble is clearly better calibrated than the MSE-ensemble even in the OoD case.

In the examples throughout the paper we chose to train models on summer data and test them on winter data. The opposite case was observed by us to be less interesting since the domain shift seemed to affect the prediction results in a milder manner, such that the OoD effect was less pronounced.

3.2. Likelihood-based UQ Assessment

Another way to assess the usefulness of the different UQ methods is using an NLL-like score (Tran et al., 2020) on a test set. Every UQ method is used to estimate a probability distribution at each point t , which we approximate to be Gaussian and denote with $N(\tilde{\mu}_t, \tilde{\sigma}_t^2)$. We emphasize the distinction of the estimated mean $\tilde{\mu}_t$ and uncertainty $\tilde{\sigma}_t$ from the mean $\hat{\mu}_t$ and variance $\hat{\sigma}_t^2$ predicted directly by a probabilistic model as the two network outputs (in case of an NLL loss function). Whereas the latter are used to define the conventional NLL loss, we use the former in order to define an NLL-like score that quantifies the usefulness of the uncertainty measure of the method and denote it by NLL_{UQ} :

$$NLL_{UQ}(t) = -\log P(y_t | N(\tilde{\mu}_t, \tilde{\sigma}_t^2)) \quad (7)$$

For each UQ method we plug in the definitions of $\tilde{\mu}_t$ and $\tilde{\sigma}_t^2$, either from Eqn. 3 or from Eqn. 4.

The NLL_{UQ} measure is influenced by the predictive accuracy as well as the quality of its UQ (Tran et al., 2020). For a given test set, a lower NLL_{UQ} value indicates a better combination of prediction accuracy and reliable uncertainty quantification.

Figure 2 compares the uncertainty measures in terms of their NLL_{UQ} score on test data. Panels (a) and (b) display the empirical distribution of the scores over the test period. In panel (a) the test period is a full year of 10-minute resolution SCADA data from the wind turbine, whereas in panel (b) the test data are 3 months of winter data. In all regimes we selected for this evaluation training and test data without anomalies (healthy data). In each of these panels 4 distributions are displayed: for each of the UQ methods, MSE-ensemble and NLL-ensemble, we train the CNN with a full year data (base) or with summer data (limited) and plot the resulting 4 distributions of the NLL_{UQ} scores for the test set. The numbers in brackets are the mean NLL_{UQ} over the test set.

The most pronounced observation from the empirical distributions is the high score peaks of the MSE-ensemble model in all 4 cases (base and limited training with both test sets). The high negative log likelihood scores are indicative of test points with true values which lie at the extreme tail of the

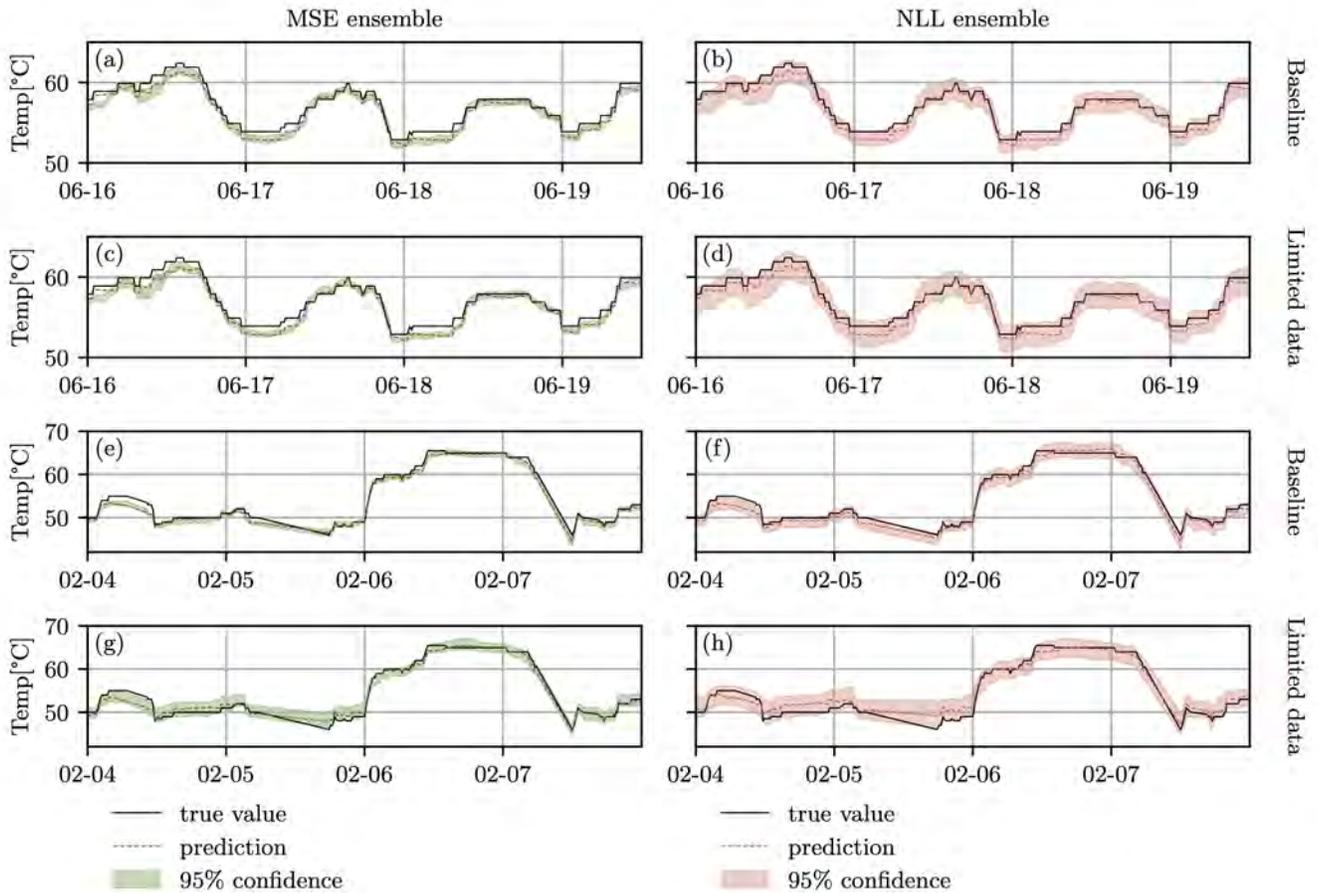


Figure 3. Confidence bands based on different uncertainty quantification (UQ) methods. Three days of the CNN ensemble predictions of the wind turbine gearbox bearing temperature (dashed) together with 95% confidence bands are compared for two different UQ methods: deep MSE ensemble (left), and deep NLL ensemble (right). The true values y_t are shown in solid lines, and the predicted mean $\hat{\mu}_t$ as dashed lines. Whereas for the MSE ensemble, true values often lie outside the predicted confidence band, the NLL ensemble clearly provides more reliable UQ, for which around 95% of the measured values lie within the predicted 95% confidence band. This improved calibration of the NLL ensemble is seen also for the case of limited training data (2nd and 4th row) and not only with a baseline of full year training data (1st and 3rd row). The two upper rows (a)-(d) show an example from summer (similar to the training conditions) whereas the two lower rows (e)-(h) show an example from winter (OoD). In both cases we chose time slots of normal (healthy) turbine conditions.

predicted uncertainty distribution $N(\tilde{\mu}_t, \tilde{\sigma}_t)$. In these cases the uncertainty estimated by $\tilde{\sigma}_t$ is too small to explain the measured value y_t given the estimated predicted value $\tilde{\mu}_t$. In other words, the peaks result from test points with strongly over-confident predictions. The over-confident predictions are characteristic of the MSE-ensemble based UQ in an "in distribution" scenario, depicted in blue (base MSE) in panel (a) (that is, when both the training and the test data cover a full year). However, this is also the case "out of distribution", when the model is trained with summer data and tested in winter (green distribution in panel (b)).

As opposed to the MSE ensemble, the UQ based on the NLL-ensemble does not suffer from strongly over-confident predictions that lead to the high NLL_{UQ} peaks. The advantage of the NLL ensemble for UQ is also evident through the lower mean NLL_{UQ} values (in brackets), both in distribution (panel (a) base models) and out of distribution (panel (b) limited models). We stress again that the term "out of distribution" is used here to describe normal (not anomalous) regimes which are not observed during training.

A direct comparison between the two UQ models is demonstrated in Figure 2(c) and (d). The NLL_{UQ} score is plotted against time for a period of 3 days using the MSE and NLL models trained with 3 months of summer data. We note that the capped values in the plots results from a regularization constant to avoid exploding logarithms. The NLL ensemble model reaches considerably lower scores as it does not suffer from over confident predictions.

This fact is visualized clearly in Figure 3. Here the 95% confidence bands around the predicted values (dashed lines) are contrasted with the true values (solid lines) of the gearbox bearing temperature of the turbine. The left and right columns of plots display the results using the MSE-ensemble and NLL-ensemble based UQ respectively, with panels (a)-(d) showing summer test data and panels (e)-(h) showing winter test data for both baseline (1 year) and limited data (3 summer months) training. Here it is clearly seen that the MSE-ensemble is strongly over confident in all regimes except OoD (panel (g)), where it is only lightly over-confident. Over-confident behaviour is easy to identify whenever the true values lie considerably outside the 95% confidence band. In a calibrated model this is expected to happen approximately 95% of the time. However, the MSE-ensemble model suffers from this considerably more often. In contrast to this, the NLL-ensemble method (right column) demonstrates almost no cases of true values well outside the confidence band, which is consistent with our observation that this model is well calibrated. The sharpness of the UQ of this model can also be observed here: the predicted uncertainty is repeatedly higher in periods of high prediction errors and lower in periods of low prediction errors.

After having demonstrated the high calibration level of the

NLL-ensemble UQ, in the next section we use this UQ method to derive an uncertainty-informed anomaly score for the fault detection task.

4. UNCERTAINTY INFORMED ANOMALY SCORE

In order to benefit from UQ for more accurate and robust anomaly detection, we suggest to incorporate the uncertainty information inside the anomaly score assigned to every prediction. In this way, the anomaly score is not based on the prediction residual alone, but takes into account the confidence (or uncertainty) of the prediction when assigning an anomaly score to a point. As a natural extension of the conventional anomaly score we described in Section 2, we define the uncertainty-informed (UI) score at step t to be the predicted CDF, evaluated at the true value y_t ,

$$S_t^{(UI)} = F(y_t; \tilde{\mu}_t, \tilde{\sigma}_t). \quad (8)$$

where $\tilde{\mu}_t$ and $\tilde{\sigma}_t$ depend on the selected UQ method. In this case, as shown in Section 3, the NLL ensemble model provides a calibrated UQ. We thus use Eqns. 4 to calculate $\tilde{\mu}_t$ and $\tilde{\sigma}_t$ for the anomaly score $S_t^{(UI)}$. Here, as well, the score is bounded between 0 and 1, and the higher it is, the more likely it is to indicate an anomaly. The threshold can be set similarly to the conventional score, in terms of the parameter α , where $S_t^{(UI)} > 1 - \alpha$ is detected as an anomaly (a fault).

In the following we compare the performance of two anomaly scores; the conventional score of Eqn. 1 and the uncertainty informed anomaly score of Eqn. 8.

Figure 4 compares the different scores as function of time for fault detection in the gearbox bearing temperature of a wind turbine. To elucidate the effect of UQ, we compare the performance using three anomaly scores: (i) the standard score $S_t^{(0)}$ of Eqn. 1 using an ensemble mean predictions of MSE-based CNNs (ii) a score based on the aleatoric uncertainty:

$$S_t^{(alea)} = F(y_t; \hat{\mu}_t, \hat{\sigma}_t). \quad (9)$$

(iii) the uncertainty-informed score $S_t^{(UI)}$ of Eqn. 8.

Every panel in Figure 4 displays the prediction residuals of the gearbox bearing temperature as function of time. Each point is colored according to its anomaly score, where blue indicates normal and red faulty given a detection threshold. In this example the significance threshold for fault detection was set on $\alpha = 0.0001$ for all methods. Panels (a)-(c) show the results achieved with a training set of a full year (marked in green shade). Panels (d)-(f) were trained with summer data only. As a result we observe a strong seasonality of the residuals, which tend to be considerably higher in winter, that is OoD, even in the absence of true faults. Panels (a) and (d) display the results of the MSE ensemble using the standard anomaly score $S_t^{(0)}$ derived using the training distribution of

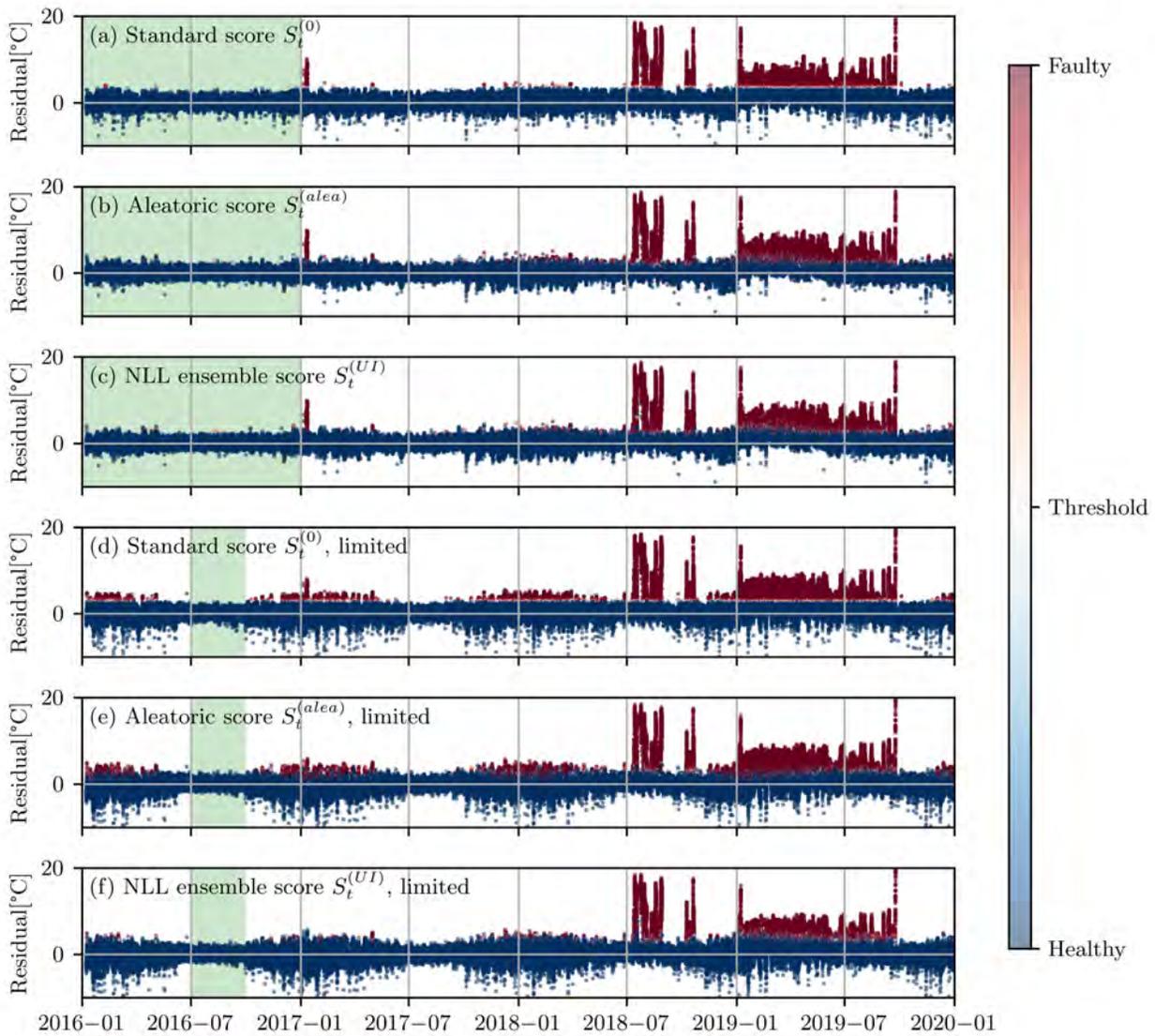


Figure 4. Comparing anomaly scores with and without uncertainty information. The prediction residuals of the gearbox bearing temperature are plotted during 4 years. In panels (a)-(c) the first year was used to train the CNN, whereas in panels (d)-(f) only three summer months were used for training, leading to strongly periodic residuals. The training period is shaded green. Three types of anomaly scores are compared: the standard score (a and d), the aleatoric score (b and e) and the NLL ensemble score (c and f). The same significance threshold $\alpha = 0.0001$ was used for all plots.

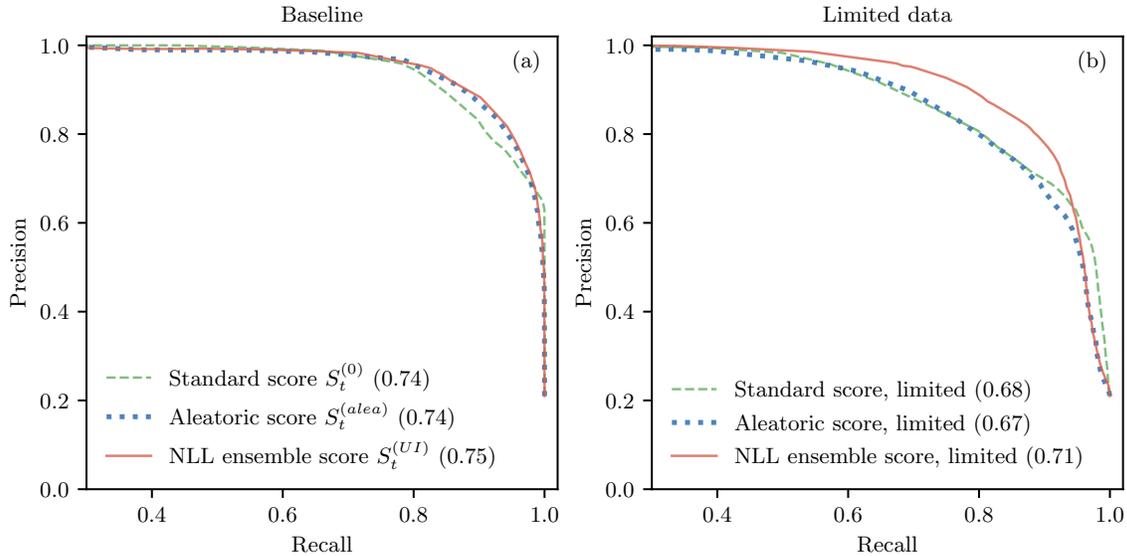


Figure 5. Fault detection performance of anomaly scores with and without uncertainty information. Three different anomaly scores are compared for two training sets: (a) one year training data (b) training data limited to 3 months from summer only. In the latter case, the test data includes winter data, which is outside the training distribution, both in healthy and in faulty conditions.

the ensemble mean residuals (see Eqn. 1). In panels (b) and (e) the residuals are the NLL model residuals based on a single realization from the ensemble. The anomaly score is the aleatoric score $S_t^{(alea)}$. In panels (c) and (f) the residuals are the ensemble mean prediction errors of the NLL model and the anomaly score is the uncertainty informed score $S_t^{(UI)}$ of Eqn.8.

It is evident that in the baseline scenario of panels (a)-(c), where the models were trained with a full year data, representing all operational conditions, the differences in performance of the three anomaly scores are small. However, when we test the scores under the OoD scenario, where only partial data was used for training we realize the need to compensate for the biased model residuals during normal periods out of distribution (i.e not during summer). The models are prone to high residuals during winter which often lead to false positives (red points) under normal conditions. Such false alarms should be avoided, as they can lead to high costs related to unnecessary downtimes and maintenance. The majority of false positives OoD is indeed avoided when the uncertainty-informed anomaly score $S_t^{(UI)}$ is used (panel (f)). OoD predictions are typically characterized by a high prediction uncertainty, and thus a wide predictive distribution. They are detected as anomalies only if their residual is large enough to reach the tail of the distribution. In this way, most of the false positives of the standard anomaly score (panel (d)) are avoided if we use the uncertainty informed score of panel (f). As expected, the score $S_t^{(alea)}$, based on the aleatoric

part of the uncertainty only, does not assess the epistemic uncertainty, and thus provides over-confident predictions OoD whose distribution is not wide enough to avoid the false positives in winter.

In order to quantify the performance of the different anomaly scores irrespective of a specific threshold, we plot their precision recall curves in Figure 5. In the absence of true normal/abnormal labels we use the baseline MSE-ensemble model as a reference, and assign the label "faulty" to predictions with an ensemble mean residual above the 95%. We observe that even with this bias in favour of the MSE-ensemble model, the NLL-ensemble score outperforms it in its fault detection fidelity.

Figure 5(a) displays the precision-recall curves of the three anomaly scores for the "in distribution" scenario, in which training data from a full year was used. In this case the performance of all scores is similar, with a slight advantage for the uncertainty-informed methods. As expected, in distribution the aleatoric score $S_t^{(alea)}$ and the fully informed score $S_t^{(UI)}$ are very similar, with similar average precision (AP) shown in brackets for each score type. On the other hand, panel (b) represents the performance OoD, since the training data is limited to normal summer data only whereas the test data includes data from the entire year. In this case, there is a clear advantage to the NLL-ensemble score (solid red), with $AP = 0.71$ vs. $AP = 0.68$ of the standard score (dashed green). The aleatoric uncertainty informed score (blue dotted) is clearly under-performing OoD, as the epistemic part

of the uncertainty is crucial in this regime.

In summary, uncertainty quantification of the DL prediction model enables us to derive an uncertainty-informed anomaly score and assign it to each new prediction. The new score outperforms the conventional anomaly score based on the entire training distribution. The advantage is more pronounced for unknown test data, which lies outside the training distribution. In case the test data is normal, the uncertainty informed anomaly score accounts for the high uncertainty in this regime and thus avoids assigning false positives, as opposed to the conventional anomaly score.

5. CONCLUSION

In this paper we introduced an uncertainty informed anomaly score, which combines the information about the prediction residual together with the prediction uncertainty into a single scalar score assigned to each prediction. The uncertainty quantification is derived using a deep ensemble of probabilistic CNNs. We demonstrated the usefulness of the uncertainty-informed score for time series anomaly detection for wind-turbine condition-based maintenance, and showed its high performance compared to conventional uncertainty-agnostic anomaly scores. The advantage is particularly clear under a distribution shift of the healthy test data with respect to the healthy training set. This situation is quite common in PHM applications, where the training data often covers only part of the normal operating conditions expected during deployment. Thus, an approach that can reduce the false alarm rate in these cases is of high relevance. However, since our approach is generic, it can be applied to anomaly detection models trained with healthy (normal) data in any application field and is not limited to time series nor to PHM applications. A central advantage of the uncertainty-informed score is that a health index can be assigned at each and every time step. This is in contrast to the common uncertainty-aware classification methods that suggest to discard high-uncertainty predictions altogether. We believe that this approach provides a systematic and transparent way to include uncertainty in deep learning algorithms for anomaly detection, increasing their reliability in practical applications.

ACKNOWLEDGMENT

The authors would like to thank Beate Sick for her useful advice. The research was funded by Innosuisse - Swiss Innovation Agency under grant No. 32513.1 IP-ICT.

REFERENCES

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., ... others (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fu-*

- sion*, 76, 243–297.
- Biggio, L., Wieland, A., Chao, M. A., Kastanis, I., & Fink, O. (2021). Uncertainty-aware prognosis via deep gaussian process. *IEEE Access*, 9, 123517–123527.
- Cai, M., Lu, F., & Sato, Y. (2020). Generalizing hand segmentation in egocentric videos with uncertainty-guided model adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14392–14401).
- Clifton, D. A., Tarassenko, L., McGrogan, N., King, D., King, S., & Anuzis, P. (2008). Bayesian extreme value statistics for novelty detection in gas-turbine engines. In *2008 IEEE Aerospace Conference* (pp. 1–11).
- Dürr, O., Sick, B., & Murina, E. (2020). *Probabilistic deep learning: With python, keras and tensorflow probability*. Manning Publications.
- Fink, O., Wang, Q., Svensen, M., Dersin, P., Lee, W.-J., & Ducoffe, M. (2020). Potential, challenges and future directions for deep learning in prognostics and health management applications. *Engineering Applications of Artificial Intelligence*, 92, 103678.
- Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning* (pp. 1050–1059).
- Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., ... others (2021). A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*.
- He, Y., Zhu, C., Wang, J., Savvides, M., & Zhang, X. (2019). Bounding box regression with uncertainty for accurate object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 2888–2897).
- Herzog, L., Murina, E., Dürr, O., Wegener, S., & Sick, B. (2020). Integrating uncertainty in deep neural networks for mri based stroke analysis. *Medical Image Analysis*, 65, 101790.
- Kraus, F., & Dietmayer, K. (2019). Uncertainty estimation in one-stage object detection. In *2019 IEEE intelligent transportation systems conference (itsc)* (pp. 53–60).
- Kuleshov, V., Fenner, N., & Ermon, S. (2018). Accurate uncertainties for deep learning using calibrated regression. In *International conference on machine learning* (pp. 2796–2804).
- Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.
- Laptev, N., Yosinski, J., Li, L. E., & Smyl, S. (2017). Time-series extreme event forecasting with neural networks at uber. In *International conference on machine learning* (Vol. 34, pp. 1–5).
- Leibig, C., Allken, V., Ayhan, M. S., Berens, P., & Wahl,

- S. (2017). Leveraging uncertainty information from deep neural networks for disease detection. *Scientific reports*, 7(1), 1–14.
- Levi, D., Gispán, L., Giladi, N., & Fetaya, E. (2019). Evaluating and calibrating uncertainty prediction in regression tasks. *arXiv preprint arXiv:1905.11659*.
- Miller, D., Dayoub, F., Milford, M., & Sünderhauf, N. (2019). Evaluating merging strategies for sampling-based uncertainty techniques in object detection. In *2019 international conference on robotics and automation (icra)* (pp. 2348–2354).
- Sato, K., Hama, K., Matsubara, T., & Uehara, K. (2019). Predictable uncertainty-aware unsupervised deep anomaly segmentation. In *2019 international joint conference on neural networks (ijcnn)* (pp. 1–7).
- Schwaiger, A., Sinhamahapatra, P., Gansloser, J., & Roscher, K. (2020). Is uncertainty quantification in deep learning sufficient for out-of-distribution detection? In *Aisafety@ ijcai*.
- Seeböck, P., Orlando, J. I., Schlegl, T., Waldstein, S. M., Bogunović, H., Klimescha, S., ... Schmidt-Erfurth, U. (2019). Exploiting epistemic uncertainty of anatomy segmentation for anomaly detection in retinal oct. *IEEE transactions on medical imaging*, 39(1), 87–98.
- Tautz-Weinert, J., & Watson, S. (2016). Using scada data for wind turbine condition monitoring—a review. *IET Renewable Power Generation*, 11(4), 382–394.
- Tran, K., Neiswanger, W., Yoon, J., Zhang, Q., Xing, E., & Ulissi, Z. W. (2020). Methods for comparing uncertainty quantifications for material property predictions. *Machine Learning: Science and Technology*, 1(2), 025006.
- Ulmer, M., Jarlskog, E., Pizza, G., Manninen, J., & Goren Huber, L. (2020). Early fault detection based on wind turbine scada data using convolutional neural networks. In *Proceedings of the european conference of the phm society* (Vol. 5).

A Hierarchical XGBoost Early Detection Method for Quality and Productivity Improvement of Electronics Manufacturing Systems

Alexandre Gaffet^{1,2}, Nathalie Barbosa Roa¹, Pauline Ribot^{2,3}, Elodie Chanthery^{2,4} and Christophe Merle¹

¹ *Vitesco Technologies France SAS 44, Avenue du Général de Croutte, F-31100 Toulouse, France*
alexandre.gaffet@vitesco.com
nathalie.barbosa.roa@vitesco.com
christophe.merle@vitesco.com

² *CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France*
elodie.chanthery@laas.fr
pauline.ribot@laas.fr

³ *Univ. de Toulouse, UPS, LAAS, F-31400 Toulouse, France*

⁴ *Univ. de Toulouse, INSA, LAAS, F-31400 Toulouse, France*

ABSTRACT

This paper presents XGBoost classifier-based methods to solve three tasks proposed by the European Prognostics and Health Management Society (PHME) 2022 conference. These tasks are based on real data from a Surface Mount Technologies line. Each of these tasks aims to improve the efficiency of the Printed Circuit Board (PCB) manufacturing process, facilitate the operator's work and minimize the cases of manual intervention. Due to the structured nature of the problems proposed for each task, an XGBoost method based on encoding and feature engineering is proposed. The proposed methods utilise the fusion of test values and system characteristics extracted from two different testing equipment of the Surface Mount Technologies lines. This work also explores the problems of generalising prediction at the system level using information from the subsystem data. For this particular industrial case: the challenges with the changes in the number of subsystems. For Industry 4.0, the need for interpretability is very important. This is why the results of the models are analysed using Shapley values. With the proposed method, our team took the first place, capable of successfully detecting at an early stage the defective components for tasks 2 and 3.

Alexandre Gaffet et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. INTRODUCTION

The 2022 PHME Data challenge encourages participants to solve multiple classification problems for a real production line from Bitron Spa. The dataset includes data from Solder Paste Inspection (SPI) and Automatic Optical Inspection (AOI) equipment of a real industrial production line equipped with automated, integrated and fully connected machines (Industry 4.0). A detailed description of the dataset is given in 3. The challenge is to design an algorithm to predict test labels for the components. Specifically, the goal is to develop a hierarchical classification predicting: 1. whether the AOI classifies the component as defective; 2. in the case of a defect, the label applied by the operator; 3. in the case of confirmation of the defect by the operator, the repair label.

To tackle this challenge, we pursue the following steps: data exploration and domain knowledge extraction, data cleaning, data preparation (normalization and encoding), data modelling (model training and validation) and results in analysis. The four last steps were made recursively while trying different approaches, as shown in Section 4. Data exploration allowed us to identify three main issues within the given dataset: missing information, highly imbalanced classes (for all tasks) and high cardinality on the categorical features. The latter is not necessarily an issue but implies that a special treatment needs to be done to these features *a priori*. We will elaborate on the issues in Section 4.1 and on the categorical encoding in Section 4.2.

To solve each task, we take different information units formed

by feature tuples, corresponding to different levels on the data hierarchy. At the same time, different features are kept as relevant for each task and followed by specific normalization or encoding. The specific tools used for each task are described in detail in Section 4.3. Finally, after presenting the experimental setup used to tune the model hyperparameters (Section 4.4), the achieved results are discussed in Section 5.

2. RELATED WORK

Several scientific articles already present machine learning applications for Surface Mount Technology production lines. (Richter, Streitferdt, & Rozova, 2017) proposes a convolution neural networks deep learning application working on the AOI system to automatically detect defects. In (Tavakolizadeh, Soto, Gyulai, & Beecks, 2017), some binary classifiers are tested to detect defects inside products using simulated production data. These data are simulated from several SMT lines and give a good classification score. In (Parvizioman, Cao, Yang, Park, & Won, 2019), a component shift prediction method is proposed to predict the shift of the pad during the reflow process. In (Park, Yoo, Kim, Lee, & Kim, 2020) SPI data are used to predict at an early stage the potential defects of the prediction. This work is based on a dual-level defect detection method. In (Jabbar et al., 2018) some tree-based machine learning methods are used to predict the defects found in AOI using SPI data. (Gaffet, Ribot, Chanthery, Roa, & Merle, 2021) proposes an unsupervised univariate method for monitoring the In-Circuit Testing machine (located at the end of the Surface Mount Technology lines) and components. Another large topic of interest for this study is prognosis and health management at different levels: system-level or sub-system level. In our case, we have to use information from a pin level to retrieve the health at a system level which is the product component. This topic is linked to the decentralized diagnosis approach. (Zhang, 2010) proposes a decentralized model-based approach with a simulation example of automated highway systems. (Ferdowsi, Raja, & Jagannathan, 2012) proposes a decentralized fault diagnosis and prognosis methodology for large-scale systems adapted for aircraft, trains, automobiles, power plants and chemical plants applications. (Tamssaouet, Nguyen, Medjaher, & Orchard, 2021) proposes a component interaction-based method to provide the prognosis of multi sub-system model.

3. DATA DESCRIPTION

The PHME provides the dataset used in this article as part of the 2022 conference data challenge (PHM Society, 2022). The dataset includes measurement information from two different steps of the PCB production (see Figure 1). The first step is the SPI, in which each solder pad is checked to verify its compliance and, accordingly, a sanction is generated depending on the quality of the solder (evaluated on several physical aspects). The second step is the AOI. In addition to

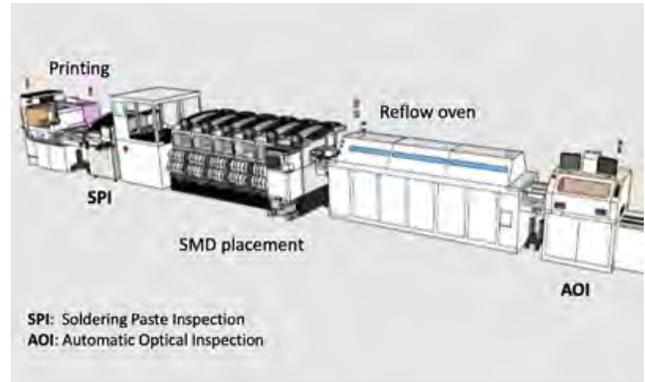


Figure 1. Surface Mount Technologies Production Line. (PHM Society, 2022)

checking the finished solder pads, their position, shape, etc., this process also inspects the component itself, looking for defects like missing or misaligned components. The AOI inspection has two types of sanctions: the automatic sanction provided by the machine itself and the one given by an operator that verifies the first one in case of spotted defects.

Table 1. Summary characteristics of the datasets

Feature	SPI	AOI
Number of lines	5 985 382	31 617
Number of panels	1924	1 924
Number of components	129	102
Figures/panel	{1,8}	{1,...,8}
Components/panel	{128,129}	{2,...,27}
Lines/panel	{3112,...}	{2,...,203}

Simple data exploration allows to discover anomalous entries and clean the datasets. A summary of the information found in each dataset is shown in Table 1. The cleaned SPI dataset is composed of 1921 panels, each with 3112 entries. A panel is an ensemble of 8 PCBs, also called figures, grouped. Each figure is composed of 129 components. The component reference is found on the feature `ComponentID`. These components have several pins that can be identified using the `PinNumber` feature. The SPI test results provide the volume, area, height, size, shape and offset of each `PadID` as well as a final result flag. A `PadID` corresponds to a unique combination of `{FigureID, ComponentID, PinNumber}`. Each panel provided by the competition has at least one component detected as a defect by AOI automatic sanction. Each line of the datasets describes a `PadID` of one electronic board.

For the AOI, each line corresponds to a unique entry of the set `{PanelID, FigureID, MachineID, ComponentID, PinNumber}`, where the `PinNumber` can be fill as `NaN`. On such cases, we believe the AOI does not inspect the solder paste but the component itself.

As introduced before, the challenge is divided into three tasks.

Task 1 is to predict whether or not a component will be classified as defective by the AOI using only the inputs provided by the SPI, i.e. the pad measurements. Task 2 is about predicting the operator’s label using the SPI test results and the automatic defect classification provided by the AOI (`AOIlabel`). Finally, task 3 concerns the reparation operation. Again, the objective is to predict whether the component detected as a defect by the operator can be repaired or not. For this task, the information used for the prediction is the SPI test result, the `AOIlabel` and also the `OperatorLabel`.

4. METHODOLOGIES

4.1. Challenges

The exploratory analysis of the training data has revealed several issues that need to be tackled to solve correctly the different tasks:

1. Missing values: for three different `PanelID`, the proposed data missed some information in the SPI dataset. We choose to exclude the lines with no information.
2. Class imbalance: the number of components detected as defects by the AOI is much lower than the number of components from the SPI dataset. Similarly, the number of components really classified as a fault by the operator is much lower than the number of components classified as correct.
3. High cardinality of the categorical features: the categorical features `PadID` have more than 1000 modes. Without any sort of variable encoding, classifiers are very difficult to use for such variables. `PadID` is already encoded in the sort because the values are ordered by `FigureID`. In a way, the variable is encoded by component area.
4. High bias in continuous features: the continuous features such as volume, area, height... are highly correlated to the categorical feature `PadID`.
5. Level of prediction: the prediction has to be done at a component level, whereas the available data are given at a `PadID`. This leads to a lot of issues in creating the target training and prediction. Indeed, for instance, it is unclear if the target training has to be created by component or by pad.
6. Different numbers of pins: the number of pins depends on the component. It is difficult to use all the pin results as input of a classifier for each component, as the number of pins and therefore features varies. The generalisation of the training depending on the component is very difficult.

4.2. XGBoost algorithm and categorical features encoding

For tabular data applications, Gradient Boosting Decision Trees (GBDT) are widely used, being XGBoost (Chen et al., 2015),

CatBoost (Prokhorenkova, Gusev, Vorobev, Dorogush, & Gulin, 2018) and LightGBM (Ke et al., 2017) the algorithms with the best results. Among these algorithms, we decide to use XGBoost, which is a scalable, paralleled and distributed implementation of the original gradient boosting tree algorithm. GBDT is an ensemble model algorithm, i.e. it combines several decision trees to perform a better prediction than a single model. XGBoost uses, in particular, the idea of boosting: it uses a collection of weak models to generate a strong model. In practice, for XGBoost, the idea is to use a gradient descent algorithm over a cost function to iteratively generate and improve weak models. On each iteration, a new weak decision tree is generated based on the error residual of the previous weak model. The final prediction is a weighted sum of all the iterated weak trees. Among ensemble methods, boosting can minimize the model’s bias. We propose one model for each task. In this section, these models as well as the used features are presented.

XGBoost is a very performing algorithm, although some caution has to be taken when categorical features are used. This is true for all tree-based or boosted tree methods. In particular, one hot encoding can lead to very poor results when the categorical features have many levels. Indeed, a large number of levels leads to sparsity as for each level, a new variable is created. These new variables have only a small fraction of data points that shall have the value 1 and the other the value 0, which is a problem for tree-based methods because tree split searching for the purest nodes. Indeed, a hot encoded variable is not very likely to lead to the purest nodes if it is very sparse. That is why, the tree split will not be done using this one hot encoded variable. Even if the original categorical feature has a lot of importance for the prediction. In our cases, other encoding techniques should be used.

First, a common technique is the Hash encoding. It is already present in our dataset with a numerical value for each `PadID` level. The `PadID` level values depend on the `FigureID` of the pad. Here the Hash encoding is done with only one feature but in general, it could be encoded into more features. One of the most used hashing methods is described in (Yong-Xia & Ge, 2010).

The next approach is the frequency-based encoding method: it is based on the frequency of the levels as the label value. If the frequency is linked to the target, it will help the prediction of the variable. For instance, in tasks 2 and 3, the frequency of one component is probably linked to the issues that can exist for each component. This encoding can be useful in that case.

Finally, the last type of encoding is label-based. The idea of label encoding is to replace each categorical value with the conditional probability of the class to be predicted by knowing the categorical features. This can be done by several methods such as Leave One Out Encoding, and CatBoost en-

coding (Prokhorenkova et al., 2018). The main issue with this method is to learn the conditional probability without overfitting. It can be realized by not taking the observation into account in the learning of the probability for each observation, as in the Leave One Out Encoding. This can also be done more efficiently using CatBoost encoding. For this case, we found that the CatBoost encoding performs best.

The frequency-based encoding and Hash encoding have less success.

4.3. Solving the challenges

Task 1

The first task is the most challenging of all. The main difficulty arises from the fact that some defects are related to the pin, and others to the component itself. In our opinion, the most important question for each task is “Should we predict (model) by component or by pin?”. For this first task, we decide to go for a per-pin prediction. This is mainly guided by the difficulty to generate coherent labels and features at the component level for this task. Indeed, only one pin can have an issue. It does not seem right to affect the same label to a component with only one pin detected as a defect by AOI equipment and another component with multiple pins with defects. Moreover, the number of pins varies too much depending on the studied component. As a result, any aggregation of continuous variables will probably hide important information if only one pin has a defect. For instance, the aggregation with the mean of the `Volume` will not contain a lot of information if there are many pins, and only one defective pin for the considered component.

For each tuple, we want to predict if the tuple is detected as a defect by the AOI equipment or not. Accordingly, the training target column is 1 if the tuple appears in the AOI dataset and 0 otherwise. It is worth noting that we are not considering as defective tuples for which only `PanelID`, `FigureID`, `ComponentID` appear with `PinNumber = NaN`. Both, categorical and continuous features, are used as input. We use the following continuous variables: `Volume(%)`, `Area(%)`, `OffsetX(%)`, `OffsetY(%)`, `Shape(um)`, `PosX(mm)`, `PosY(mm)`, `SizeX`, `SizeY` that we simply call “numerical features” in the following of the article. As a categorical value, we only keep `ComponentID` that we encode using a CatBoost encoding method (Prokhorenkova et al., 2018).

Task 2

For this second task, we first split the AOI dataset into two parts according to whether or not `PinNumber = NaN`. In the case where `PinNumber` is specified, we can join the AOI and the SPI easily using the columns `PanelID`, `FigureID`, `ComponentID`, `PinNumber` in each dataset. From the joint dataset, we use the “numerical features” from the SPI test results as de-

finied in task 1. For the categorical features, we keep three features: `AOILabel`, `ComponentID` and `FigureID.ComponentID` where the later is the string concatenation of the variables with the same name. For these features, we use a CatBoost encoding based on the categorical encoders python library. We believe a better result can be achieved with deeper work on the optimization of the encoder hyper-parameters. Finally, we also create two new meta-features (not encoded) `Count_Pin_Component` and `Count_Pin_Figure`. These two variables are counting the number of pins detected as a defect by the AOI respectively for the component and figure of the tuple `PanelID`, `FigureID`, `ComponentID`, `PinNumber`. The XGBoost classifier algorithm classes each tuple into the class “Bad” or “Good” of the `OperatorLabel` target column.

For the AOI defect without any `PinNumber` associated, we propose to also use an ensemble of categories and continuous variables. We use the same categorical and meta features, while for the numerical features we use the mean values per component of the following variables: `Volume(%)`, `Area(%)`, `OffsetX(%)`, `OffsetY(%)` to keep the information only on the `PanelID`, `FigureID`, `ComponentID` tuple level.

Finally, we also use an XGBoost classifier algorithm to class each tuple `PanelID`, `FigureID`, `ComponentID`, `PinNumber`, with `PinNumber` referenced as `NaN` as described before.

For the final sanction (that must be given at the `PanelID`, `FigureID`, `ComponentID` three-tuple level), we use the following rule: if one of the four-tuple `PanelID`, `FigureID`, `ComponentID`, `PinNumber` entry is predicted as “Bad”, then the associated three-tuple will also be considered as “Bad”. If not, the label “Good” is assigned to the three-tuple.

Task 3

Task 3 is about the prediction of one categorical value presented in the AOI dataset `RepairLabel`. This label can take two values, `FalseScrap` or `NotPossibleToRepair`. For this task, we tried the same approach as in task 2 predicting each four-tuple `PanelID`, `FigureID`, `ComponentID`, `PinNumber` and merging the results per component, but the approach was not successful. To improve the result, we choose to do the prediction for the `PanelID`, `FigureID`, `ComponentID` three-tuple directly. As input, we use the same method as for task 2, grouping the SPI values per three-tuple using the mean as an aggregation method for the “numerical features”. The categorical variables used are `ComponentID`, `FigureID.ComponentID`. As before, these variables were encoded using the CatBoost encoding method.

The meta-features generated are `Count_Pin_Component`, `Count_Pin_Figure` and `Count_Pin_Panel`, respectively, the number of pins detected as defects by the AOI per component, figure and panel. We also created one-hot encoded features from the `AOILabel`. For each labelled type of error, the asso-

ciated feature has the value of 1 if the three-tuple is detected as having this error by the AOI machine (in at least one pin) and 0 otherwise. To predict the class of each tuple, we use the XGBoost classifier.

4.4. Hyper-parameter tuning

The XGBoost model has been tuned using the Optuna python library (Akiba, Sano, Yanase, Ohta, & Koyama, 2019). This package is an optimization framework that searches for the best hyper-parameters of the space defined by the user. It uses some distributed computation and early stopping to improve the speed of the solution. It is implemented with various optimization algorithms. In our case, we use the sampling algorithm Tree Parzen Estimator Sampler (TPES) (Bergstra, Bardenet, Bengio, & Kégl, 2011). It is a sequential model-based optimization. As a bayesian optimization algorithm, it computes a probability model of the optimization function and selects the best hyper-parameter according to this probability model and the real cost result. For each step, the tuning is done using the mean of the F1-score of a 4 cross-validation method. The dataset is split into four parts, each of these parts is recursively the testing dataset while the other is used for training the model. For the sake of reproducibility, hyper-optimization techniques and training algorithms are available in (Gaffet., 2022). Our prediction can probably be easily improved with more iterations of the tuning phase, as we did not spend much time on it.

5. RESULTS AND DISCUSSION

The results obtained for the three tasks are detailed in this section.

5.1. Score

Table 2. F1-score for the three tasks with training and testing data set

Dataset	Task 1	Task 2	Task 3	Score
Training	0.43	0.66	0.90	0.66
Testing	0.41	0.67	0.77	0.62

Table 2 shows the F1-score for each task of the challenge. The training and testing set results are close, showing good generalization capability. It seems that our models avoid overfitting issues that are difficult to handle with this dataset. Indeed, the imbalance issue and the fact that some variables such as ComponentID have a lot of importance can lead to large bias.

5.2. Feature importance

Figure 2 presents the feature importance of the classification model for task 1. The most important features are the component’s position on the panel and the encoded ComponentID variable. Because almost 30 per cent of the defects found in the AOI dataset are coming from one component (“BC1”),

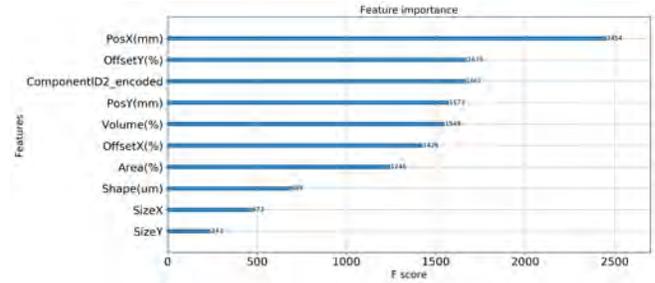


Figure 2. Task 1: feature importance of the XGBoost classifier

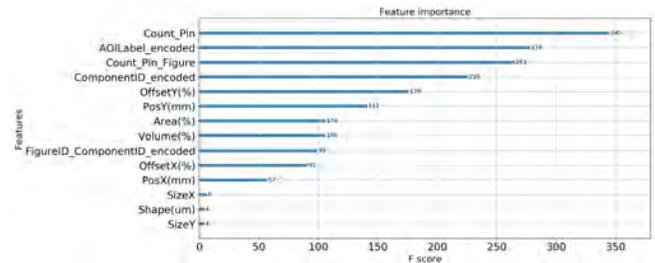


Figure 3. Task 2: feature importance of the XGBoost classifier for the four-tuple (AOI defect is linked to a pin)

this creates a bias in the model results that depends on ComponentID features.

Figure 3 and Figure 4 present the feature importance for task 2 on the four-tuple and three-tuple classifiers respectively. The position and the ComponentID are also important for this task, but the continuous variables Volume(%) , OffsetX(%) , OffsetY(%) and Shape(um) seem to have also a great impact on the prediction. Even more, the generated meta-features Count_Pin_Component, representing the number of pins with a defect per component, and Count_Pin_Figure, representing the number of defective pins per figure, have also a large impact. Actually, these two features allow improving the F1-score for this task by almost 0.2.

Figure 5 shows the feature importance of the XGBoost clas-

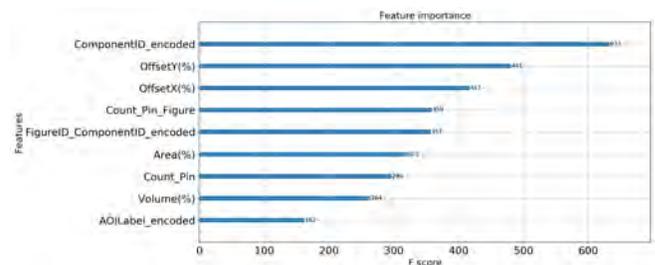


Figure 4. Task 2: feature importance of the XGBoost classifier for the three-tuple (AOI defect is not linked to a pin)

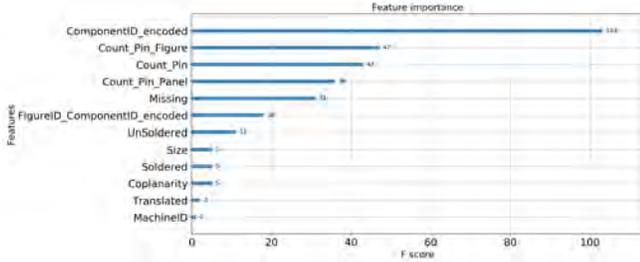


Figure 5. Task 3: feature importance of the XGBoost classifier

sifier for task 3. Without surprise, given the previous results, ComponentID is shown as the feature with the biggest importance value. ComponentID could be thought of as a domain-specific key factor, if, for example, depending on the type of component, the reparation is possible / allowed or not. For instance, if there is an issue with a microchip, this is far more difficult and costly to solve than an issue with a simple resistance. The following most important features are the meta-features being in the order of importance more critical the number of defective pins per figure, then component and final panel. This also seems correct as the number of issues increases the potential damage to the product. The “Missing” AOILabel (one-hot encoded) also has a considerable impact. Maybe it is not possible to replace a missing component due to oven operation.

5.3. SHAP values

The interpretation of the machine learning model described as a black block model is a really important topic in the industry. Indeed, for the acceptance of a model, it is mandatory to explain the model decision to the process experts. The interpretation allows validating the model by comparing the model and experts’ explanation of a phenomenon. It also allows recommending some repair actions to the experts. SHAP (SHapley Additive exPlanation) interpretation (Lundberg & Lee, 2017) is based on the game theory (Štrumbelj & Kononenko, 2014). The idea is to compute an interpretation value called SHAP value and denoted ϕ_j for all variables and each sample of the dataset. The output of the model is described by the sum of the SHAP values ϕ_j .

$$\phi_j = \sum_{S \subseteq J \setminus \{j\}} \frac{|S|!(M - |S| - 1)!(f(S \cup \{j\}) - f(S))}{M!} \quad (1)$$

where M is the number of variables, J is the ensemble of variables, f the model output and j the variable index. SHAP is an additive method. The output of a classifier can be described as the cumulative sum of the impact of all variables.

Figure 6, Figure 7, Figure 8, Figure 9 present the computed impact of the features on the model output. For task 1, a

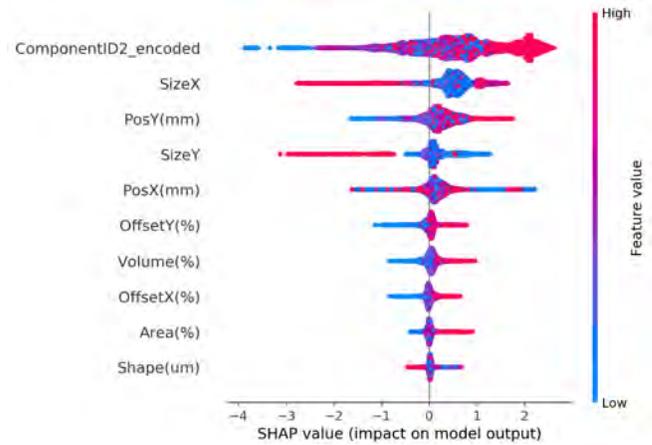


Figure 6. Task 1: SHAP value of the XGBoost classifier

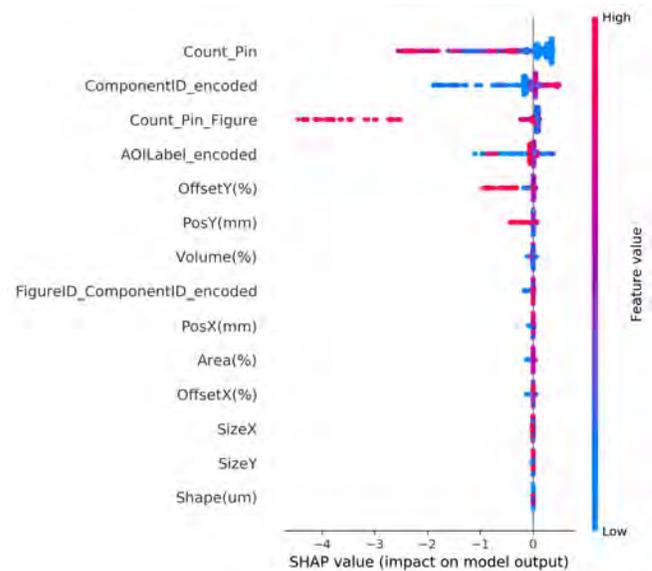


Figure 7. Task 2 for the four-tuple (AOI defect is linked to a pin): SHAP value of the XGBoost classifier

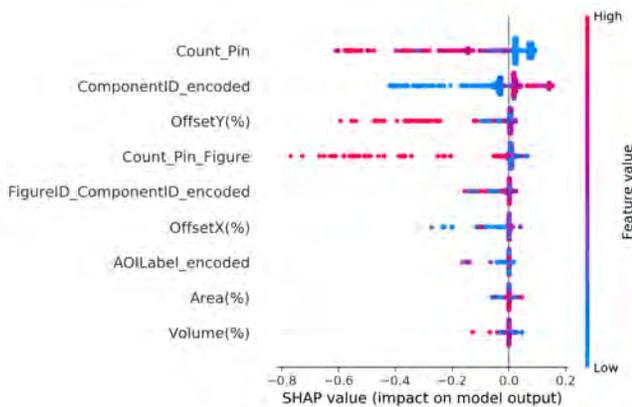


Figure 8. Task 2 for the three-tuple (AOI defect is not linked to a pin): SHAP value of the XGBoost classifier

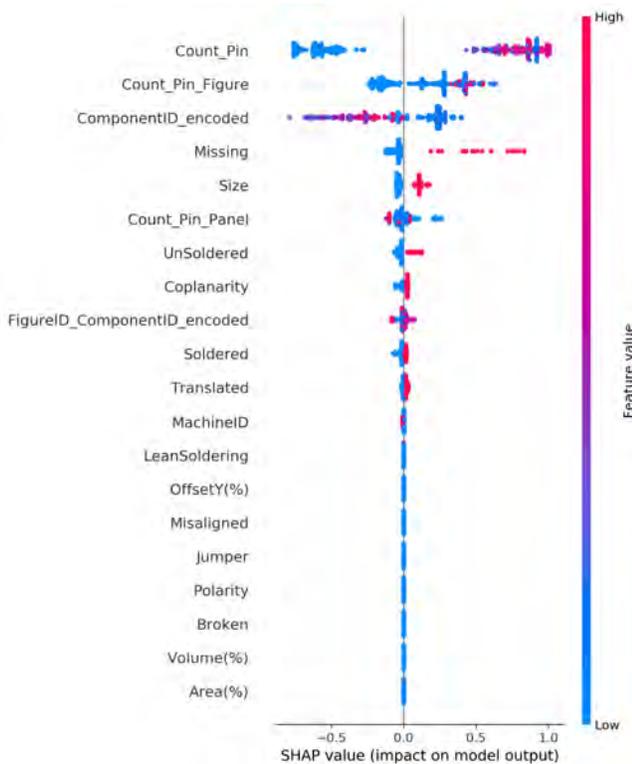


Figure 9. Task 3: SHAP value of the XGBoost classifier

		Predicted Label	
		Defects	Not Defects
True labels	Defects	8263	23345
	Not Defects	4204	1937814

Table 3. Confusion matrix for all training sets

		Predicted Label	
		Defects	Not Defects
True labels	Defects	8256	1064
	Not Defects	4203	1922

Table 4. Confusion matrix for the training set observation of the component “BC1”

larger value of the CatBoost encoded variable `ComponentID` logically results in larger model output (the encoding is done by replacing the level value with its conditional probability of target). The variable `SizeX` does not seem to be connected to the model output. A more interesting larger value of the continuous variable of `Volume` and `Offset` seems to be representative of a defect for task 1. For task 2, a low number of defective pins per component and per figure is more likely to be a “Good” `OperatorLabel`. This result was expected as the more the AOI equipment found defective pins, the more likely a real defect in the component is. The impact of continuous variables is very low for this task. For Task 3, the output value 1 is associated with the “Not Possible to Repair” label and the output value 0 with “FalseScrap”. The SHAP values indicate that the more the number of defective pins per component per figure, the more likely the product is impossible to repair. If a component or a pin is missing, then the product seems also more likely to be not possible to repair. The same result is found for “Unsoldered” `AOILabel` level. For the other level of `AOILabel`, the results are unclear.

5.4. Issues with Task 1

In this subsection, the issues encountered in the prediction of task 1 are detailed. As previously reported, the results of the prediction of task 1 are highly biased by one component “BC1”. Table 3 and Table 4 present the confusion matrix results of the training of task 1. It can be seen that most of the correctly detected defects by the classifier model of task 1 is coming from the results of the component “BC1”. Only five other tuples are correctly classified for the defects. Obviously, this is a concern for our method. Another observation is that the F1-score obtained with the results coming from Table 4 is 0.76 whereas the F1-score obtained considering all the tuples as defects is 0.75. Both scores are very close and depending on the cross-validation random selection, the simple rule considering all the tuples of the the “BC1” component as defects can be better than a more advanced classifier. To solve this issue, more advanced analyses are required. From our experience, it seems easier to predict the defects associated with the “Unsoldered” `AOILabel` than the other.

AOILabel / Score	Score Task 1
Coplanarity	0.00
Translated	0.00
Soldered	0.00
Unsoldered	0.55
Size	0.00
LeanSoldering	0.27
Misaligned	0.15
Missing	0.00
Jumper	0.00

Table 5. Task 1 score results for different AOILabel levels

6. CONCLUSION AND FUTURE WORK

To solve the different challenges in the Printed Circuit Board production, we proposed three different methods. These methods use the XGBoost classifier and are based on variable encoding and feature engineering. We have shown that generating meta-features representing the circuit state at different levels (component, figure and panel) improved the generalization of the classifiers. The use of different encoding techniques for the categorical features has shown CatBoost as the most promising one, due to the high cardinality issues. The first task results are probably too biased to be used in production. However, the results of task 2 and task 3 are promising for a production application.

In future work, it could be interesting to add some information to improve the prediction. The time between operations for instance is missing (probably to not share cycle time information). A product with a larger cycle can lead to an issue with dust deposit, humidity change... This information could improve the performance of all tasks. Some extra information like the temperature curves of the oven can also help to better predict future issues. Finally, for task 1, unsupervised monitoring methods may be a better solution due to the class imbalance issue.

REFERENCES

- Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining* (pp. 2623–2631).
- Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24.
- Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., ... others (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2, 1*(4), 1–4.
- Ferdowsi, H., Raja, D. L., & Jagannathan, S. (2012). A decentralized fault detection and prediction scheme for nonlinear interconnected continuous-time systems. In *The 2012 international joint conference on neural networks (ijcnn)* (pp. 1–7).
- Gaffet, A. (2022). *Phme data contest 2022*. <https://github.com/alexandregft/PHME-data-contest>. GitHub.
- Gaffet, A., Ribot, P., Chanthery, E., Roa, N. B., & Merle, C. (2021). Data-driven capability-based health monitoring method for automotive manufacturing. In *Phm society european conference* (Vol. 6, pp. 12–12).
- Jabbar, E., Besse, P., Loubes, J.-M., Roa, N. B., Merle, C., & Dettai, R. (2018). Supervised learning approach for surface-mount device production. In *International conference on machine learning, optimization, and data science* (pp. 254–263).
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.
- Park, J.-M., Yoo, Y.-H., Kim, U.-H., Lee, D., & Kim, J.-H. (2020). D 3 pointnet: Dual-level defect detection pointnet for solder paste printer in surface mount technology. *IEEE Access*, 8, 140310–140322.
- Parvizioman, I., Cao, S., Yang, H., Park, S., & Won, D. (2019). Data-driven prediction model of components shift during reflow process in surface mount technology. *Procedia Manufacturing*, 38, 100–107.
- PHM Society. (2022). *Data challenge*. (Sponsored by Bitron Spa. Published electronically at <https://phm-europe.org/data-challenge>)
- Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.
- Richter, J., Streitferdt, D., & Rozova, E. (2017). On the development of intelligent optical inspections. In *2017 IEEE 7th annual computing and communication workshop and conference (ccwc)* (pp. 1–6).
- Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3), 647–665.
- Tamssaouet, F., Nguyen, K. T., Medjaher, K., & Orchard, M. E. (2021). Online joint estimation and prediction for system-level prognostics under component interactions and mission profile effects. *ISA transactions*, 113, 52–63.
- Tavakolizadeh, F., Soto, J., Gyulai, D., & Beecks, C. (2017). Industry 4.0: mining physical defects in production of surface-mount devices.
- Yong-Xia, Z., & Ge, Z. (2010). Md5 research. In *2010 second international conference on multimedia and in-*

formation technology (Vol. 2, pp. 271–273).

Zhang, X. (2010). Decentralized fault detection for a class of large-scale nonlinear uncertain systems. In *Pro-*

ceedings of the 2010 american control conference (pp. 5650–5655).

Application of Machine Learning Methods to Predict the Quality of Electric Circuit Boards of a Production Line

Immo Schmidt¹, Lorenz Dingeldein², David Hünemohr³, Henrik Simon⁴, and Max Weigert⁵

^{1,2,3,4,5} *Technische Universität Darmstadt, Institute of Flight Systems and Automatic Control, Darmstadt, 64287, Germany*
{schmidt, dingeldein, huenemohr, simon, weigert}@fsr.tu-darmstadt.de

ABSTRACT

For the data challenge of the 2022 European PHM conference, data from a production line of electric circuit boards is provided to assess the quality of the produced components. The solution presented in this paper was elaborated to fulfill the data challenge objectives of predicting defects found in an automatic inspection at the end of the production line, predicting the result of a following human inspection and predicting the result of the repair of the defect components. Machine learning methods are used to accomplish the different prediction tasks. In order to build a reliable machine learning model, the steps of data preparation, feature engineering and model selection are performed. Finally, different models are chosen and implemented for the different sub-tasks. The prediction of defects in the automatic inspection is modeled with a multi-layer perceptron neural network, the prediction of human inspection is modeled using a random forest algorithm. For the prediction of human repair, a decision tree is implemented.

1. INTRODUCTION

The fourth industrial revolution leads to increasingly automated production and manufacturing. Production machines that are fully connected and fully equipped with sensors generate huge amounts of data enabling new data-driven approaches to assess the quality of the produced parts. Machine learning algorithms are used in an increasing number of applications in production, even if their use is often part of research and not yet widely spread (Mayr et al., 2019; Liukkonen, Havia, & Hiltunen, 2012).

Within the production environment, machine learning provides the opportunity to process the large amounts of data to improve quality, lower costs or increase the flexibility of the process and can contribute to sustainable manufacturing (Mayr et al., 2019; Cioffi, Travaglioni, Piscitelli, Petrillo, &

De Felice, 2020). In recent years, deep learning approaches for smart manufacturing have been increasingly studied (Wang, Ma, Zhang, Gao, & Wu, 2018). In contrast to engineering features with expert knowledge of the manufacturing process, deep learning provides an end-to-end machine learning approach, but oftentimes lacks interpretability of the results (Wang et al., 2018).

Within the framework of data challenges in the field of Prognostics and Health Management, several objectives related to various industrial use cases have already been addressed. In this context, the suitability of different algorithms could be demonstrated and compared. As a result, it has been possible to gather new knowledge about problem-specific approaches and insights into general solution strategies (Huang, Di, Jin, & Lee, 2017). Data sets from the manufacturing industry are currently scarce, but very useful for investigating data-based improvements in maintenance and quality control processes (Jourdan, Longard, Biegel, & Metternich, 2021).

Predictive quality is the main focus of the data challenge for the 7th European Conference of the Prognostics and Health Management Society 2022 that is held in cooperation with Biron Spa. The participants of the challenge receive data from a production line manufacturing electric circuit boards. In the production process, the surface mount technology (SMT) is used, which comprises of several manufacturing and inspection steps. The use of advanced data-driven algorithms for quality management in mass soldering processes dates back to the 1990s (Liukkonen et al., 2012). Since then the technology for the production of electronic components as well as the capabilities of machine learning algorithms evolved. Images of optical inspections of the circuit boards solder prints can be processed with machine vision algorithms to detect defects directly (Zakaria, Amir, Yaakob, & Nazemi, 2020). Indirectly supervised learning approaches enhance the outcome of current automatic optical inspections and measured solder joint dimensions. This can be used in the production line to support the operator in assessing defect calls from the automatic optical inspection and identify false positive classifications (Jabbar et al., 2019). Specifically, Jabbar et al. compared

Immo Schmidt et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

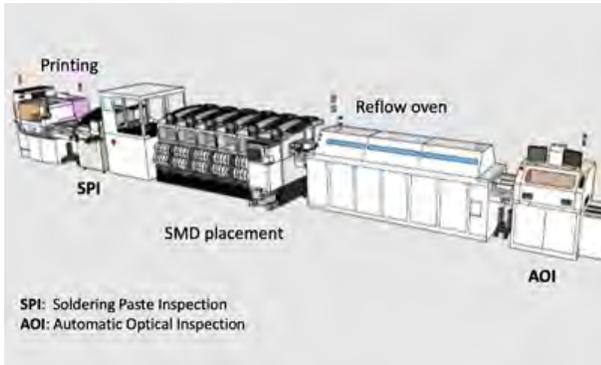


Figure 1. PCB production line (PHM Society, 2022)

tree-based machine learning algorithms for this use case and achieved a good performance (Jabbar et al., 2019). Supervised deep learning has also been applied to real production environment data to enhance the outcome of the optical inspections and reduce labor cost (Chang, Wei, Chen, & Hsieh, 2018).

1.1. Use Case

The production line for the manufacturing of electric circuit boards is depicted in Figure 1. In a first production step, a printing machine places the solder paste on the initially bare printed circuit boards (PCB). Subsequently, the electronic components are mounted on the PCB in the surface mount device (SMD) placement and the soldering process is finished in the reflow oven. Two different datasets are retrieved from the production process. The first one stems from a solder paste inspection (SPI) conducted after the placement of the solder paste and before mounting the electronic components. In this inspection, the quality of the solder paste placing in terms of, among others, volume, height and position is measured for each pin of the PCB. The second set of data is recorded in an automatic optical inspection (AOI) that follows after the soldering. During this inspection, defects on the produced PCBs shall be automatically detected. In case of defect detection, an operator visually inspects the PCB and confirms or rejects the defect found by the AOI. Only in case of a confirmation by the operator, a second operator investigates whether the defect can be repaired.

1.2. Objectives

The objective of the data challenge is the prediction of the automatic optical inspection for each component and, in case of a defect, the prediction of the assessment of the two operators. Consequently, the challenge is divided into three sub-tasks:

1. Prediction of components with a detected defect in AOI based on the SPI data. For each detected defect, a so-called *AOI Label* with information on the defect type is assigned to the component. Prediction of the defect type

is not required, the only objective is to predict whether there is an *AOI Label* for a component. The models are evaluated using the F1-score of the defect class.

2. Prediction of human inspection. In case of an AOI defect, predict whether the defect is confirmed or rejected by the human operator, that means predict the binary *Operator Label*. For this task, the SPI data and the assigned *AOI Labels* can be used as input data. The F1-score of the class of confirmed defects is used for model evaluation.
3. Prediction of Human Repair. For confirmed defects predict whether the component is false scrap or not possible to repair, that means predict the *Repair Label*. As for the prediction of human inspection, both the SPI data and the *AOI Labels* can be used as input data. The models are evaluated using the macro-averaged F1-score of the *Repair Label*.

2. APPROACH

In order to solve the given task, the cross-industry standard process for data mining CRISP-DM is followed (Chapman, P. et al., 2000). After having understood the use case defined in the section above, the next important step is to examine the given data. The SPI data set contains information on every pin of the printed circuit boards. Apart from the necessary information to identify the pin, it contains geometry data of the solder paste that is placed for soldering of the pins. This includes the measured volume, area and height of the paste. For volume and area, the data contains also percentages in relation to the target values. Furthermore, it also includes information on the shape and the target sizes of the solder paste deposit, the target position and the percentage offset in x- and y- direction. At last, there is a *SPI Result* indicating several warnings if one of the geometric values of a pin exceeds or falls below certain thresholds.

All the data from the SPI can be used as features for the tasks to solve. However, not all of the features are independent of each other. For example, the volume of the solder paste is the simple product of area and height. These relations are considered later when selecting features for the machine learning applications. Additionally, we define the percentage height in relation to the target height of the solder paste deposit as a new useful feature. It is deducted from the available geometry information as shown in Eq. 1.

$$Height(\%) = \frac{Height(\mu m)Volume(\%)Area(\mu m^2)}{Volume(\mu m^3)Area(\%)} \quad (1)$$

The AOI data set consists of all defects that are found in the automatic inspection. For every defect, the affected component is indicated and the *AOI Label* with the type of defect, *Operator Label* and *Repair Label* are given. Moreover, the affected pin of the component is given if available. This means

that some of the defects can be assigned to a specific pin while others can only be assigned on component level. It is also possible that there is more than one defect for a pin or a component. As a consequence, one step for the data preparation must be the assignment between the SPI data that are available for each pin and the AOI data that might not be assigned to a specific pin. With the three sub-tasks being independent of each other, the way of preparing the data and modeling is different and specific for each task. In the following, the chosen approach for each of the sub-tasks will be explained.

For the purpose of validation, the available data are divided into a training and a hold-out validation set. This allows the testing of models on data not used for training and the optimization of hyperparameters. To split the data, 25% of the total PCB panels are randomly chosen and form the validation set while the other 75% form the training set. After validation, the generated models are trained again on the whole data set to use all available data for learning.

2.1. Prediction of AOI defects

A first, simple idea for the prediction of defect components is that high deviations from the target values of the solder paste deposit lead to a defect warning in the AOI. As there is a *SPI Result* in the SPI data indicating exactly these high deviations, a first try is to classify all components with at least one *SPI Result* that is not "GOOD" as defect. However, this rule-based approach only leads to an F1-score of approximately 7%.

As a consequence, several more complex models are investigated. Besides decision tree and random forest classifiers, a neural network is chosen instead of the simple rule-based approach and leads to best results. The model is implemented using the Scikit-Learn library (Pedregosa, F. et al., 2011). The model is trained on pin level. That means, every pin in the SPI data is labeled depending on the presence of its component in the AOI data. The algorithm then predicts for every pin whether it is faulty. A component is classified as defect when there is at least one faulty pin.

As input features, the geometrical information on the solder paste deposit are taken from the SPI data without any further preprocessing. This includes the features "Volume(%)", "Height(μm)", "Area(%)", "OffsetX(%)", "OffsetY(%)", "SizeX", "SizeY", "Shape(μm)", "PosX(mm)" and "PosY(mm)". The neural network is a multi layer perceptron trained with 3 hidden layers consisting of 20, 50 and 10 neurons, which are the results of a small grid search. The ReLU activation function is used and alpha is set to 0.00001.

2.2. Prediction of human inspection

The human operator decides for every component with an AOI defect whether the component is "Good" or "Bad". To

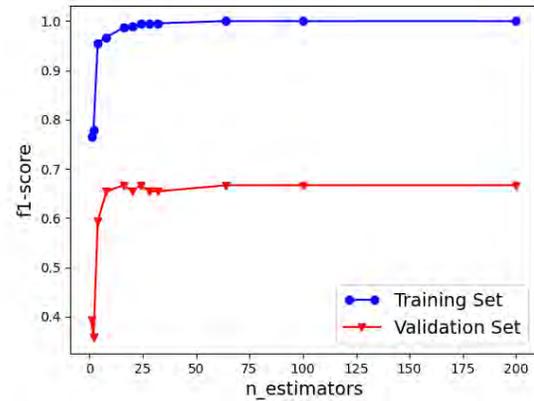


Figure 2. Optimization of the number of estimators for the prediction of human inspection.

predict the label given by the operator, we also build a model that works on component level. By consequence, the information on the pins in the SPI data and the AOI defects in the AOI data has to be aggregated for each component.

To use the categorical data from the *AOI Label*, one hot encoding is performed and the encoded labels are summed up for each component. Moreover, the total number of AOI entries is counted for each component and used as an additional feature. For the SPI data, the maximum and minimum values of the percentage area, height, offset in x- and y-direction and the shape are calculated for every component. By consequence, only the pins with the highest deviations from the target values are used for the prediction. The remaining information on the solder paste is not taken into account. Furthermore, the size of the solder paste deposit in x- and y-direction is considered as an additional feature, as percentage deviations might be more or less critical depending on the size of the pin.

With all these features, a random forest is learned on the training data to perform the binary classification. Again, the random forest algorithm is implemented using the Scikit-Learn library (Pedregosa, F. et al., 2011). To tackle the high class-imbalance with only about 1.5% of the components classified as "Bad", the class weight is set to balanced. Consequently, the "Bad" examples are weighted much higher than the "Good" examples. A hyperparameter optimization is performed using the validation set to find the optimum number of estimators and maximum depth of the trees. The ideal number of estimators is found to be around 16 as higher numbers of estimators do not lead to a higher performance. This is shown in Figure 2. The maximum depth of the trees is set to 10. Figure 3 shows that higher depths only lead to overfitting but do not significantly increase the performance on the hold out validation set.

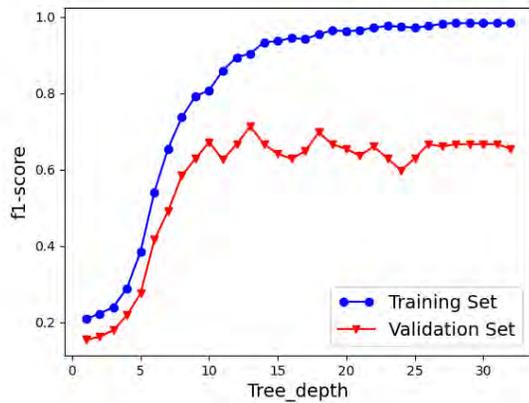


Figure 3. Optimization of the depth of the trees for the prediction of human inspection.

Table 1. Comparison of the number of AOI entries per component.

RepairLabel	Components	AOI entries	Ratio
NotPossibleToRepair	225	995	4.4
FalseScrap	122	139	1.1

2.3. Prediction of human repair

The *RepairLabel* of the second operator is only assigned to components indicated as "Bad" by the first operator. As mentioned, this concerns only about 1.5% of the components with an AOI defect. Thus, the data base used to learn the model for the prediction of human repair is rather small. The data base is even more reduced as some of the components have a *RepairLabel* set to "NotClassifiedYet" and the prediction of not classified components is not part of the task. In total, there are only 347 components left out of initially more than 27,000 components with an AOI defect.

Due to this low number of data, we tried to keep the model as simple as possible in order to reduce the risk of overfitting and guarantee the generalizability to unknown data. As shown in Table 1, components that are not possible to repair have in general much more entries in the AOI data than "FalseScrap" components. The number of AOI entries per component is on average four times higher. As a result of these considerations, we count the number of AOI entries for each component and learn a decision tree on that sole feature.

The resulting model is equivalent to a rule-based approach where all components with only a single defect entry in the AOI data are classified as "FalseScrap" and components with more than one entry in the AOI data are classified as "NotPossibleToRepair".

Table 2. Final scores of the chosen models.

Data set	Task 1 (ANN)	Task 2 (RF)	Task 3 (DT)
Training	0.39	0.81	0.85
Validation	0.39	0.66	0.87
Test	0.41	0.38	0.70

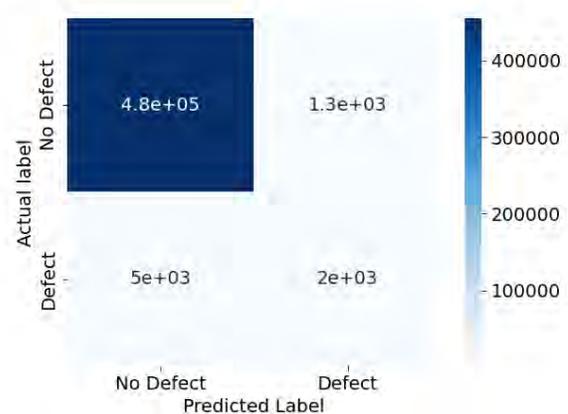


Figure 4. Confusion matrix on validation set for the prediction of AOI defects.

3. RESULTS

The results of the chosen models are presented for each sub-task separately. An overview of the scores on the different data sets is given in Table 2. The test set was not provided to the participants and only used for evaluation of the data challenge by the organizers.

3.1. Prediction of AOI defects

The neural network for predicting components with AOI defects reaches an F1-score of 0.39 on the training and validation set. A further look into the confusion matrix depicted in Figure 4 shows that the precision is at approximately 0.60 while the recall is 0.28 meaning that the model is rather weak at predicting actual components with AOI defect correctly. However, it is much better at avoiding false positives and predicting healthy components correctly.

This behaviour can be partly explained by the given class imbalance. Since there are much more components without a defect, the model is fitted to those components. Furthermore, an accurate prediction of defect components solely relying on the information on the solder paste as input might not be possible. The cause of an AOI defect can theoretically lie in a later production step like the mounting of the components for which no data are available. As shown in Table 2, the model works well on unknown data and achieves comparable scores on the test set.

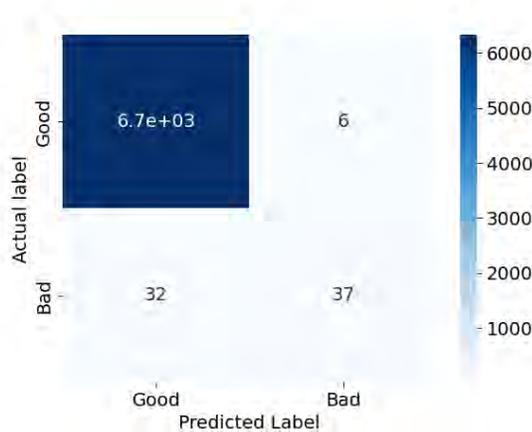


Figure 5. Confusion matrix on validation set for the prediction of human inspection.

3.2. Prediction of human inspection

For the prediction of human inspection, the generated random forest classifier reaches an F1-score of 0.81 on the training data and 0.66 on the validation data. Similar to the prediction of AOI defects, the precision of the model is higher than the recall. For the training data, the precision is 0.88 and the recall 0.73

The confusion matrix of the prediction on the validation set is shown in Figure 5. Apparently, the model precision is high even on data not used for training while the recall drops significantly. The model has no problem in predicting good components correctly. Despite balanced weights during training, it has more difficulties in predicting bad components. One reason for that might be the relatively low total number of components with a "Bad" *OperatorLabel*.

The test score only amounts to 0.38 and is thus significantly lower than the score on the validation set. This indicates that the trained and validated model tends to overfit and does not generalize well on the unknown test data.

3.3. Prediction of human repair

The decision tree shows good results on training and validation set with a combined F1-score of about 0.85. Thus, the extremely simple model seems to be a surprisingly good predictor. The confusion matrix on the validation set in Figure 6 shows that almost all of the "FalseScrap" components are correctly predicted as "FalseScrap". For the "NotPossibleToRepair" components, there are a few more falsely predicted components. However, the correct label is assigned to the prevailing majority of components.

There have been several attempts to improve the model by adding more input features. Since these attempts did not lead to a significantly better training and validation score, it was

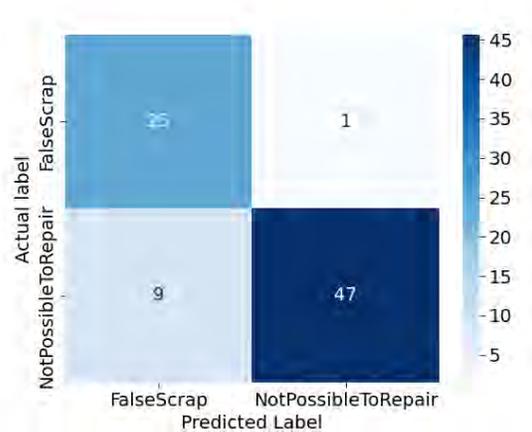


Figure 6. Confusion matrix on validation set for the prediction of human repair.

decided to stay with the simple decision tree based on the number of entries in the AOI data.

The average F1-score on the test data set is 0.70 and therefore a bit lower than the training score. It seems that the relation between the number of AOI entries and the *RepairLabel* is less evident on the test data, but the decision tree classifier still provides satisfactory results.

4. CONCLUSION

The selected models for predicting the quality of manufactured electric circuit boards show overall satisfactory results and our team was able to reach 4th place at the data challenge. However, there is still room for improvement. The chosen model for the prediction of human inspection clearly shows signs of overfitting and should be adjusted to better classify the components. Results of the prediction of human repair show that in some applications simple rule-based approaches can provide very good results comparable to those of complex machine learning models. A good understanding of the data based on an explorative data analysis is key to identify fundamental relationships in the data.

REFERENCES

- Chang, Y.-M., Wei, C.-C., Chen, J., & Hsieh, P. (2018). Classification of solder joints via automatic mistake reduction system for improvement of aoi inspection. In *2018 13th international microsystems, packaging, assembly and circuits technology conference (impact)* (pp. 150–153).
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *Crisp-dm 1.0: Step-by-step data mining guide..*
- Cioffi, R., Travaglioni, M., Piscitelli, G., Petrillo, A., &

- De Felice, F. (2020). Artificial intelligence and machine learning applications in smart production: Progress, trends, and directions. *Sustainability*, 12(2). doi: 10.3390/su12020492
- Huang, B., Di, Y., Jin, C., & Lee, J. (2017). Review of data-driven prognostics and health management techniques: Lessons learned from phm data challenge competitions..
- Jabbar, E., Besse, P., Loubes, J.-M., Roa, N. B., Merle, C., & Dettai, R. (2019). Supervised learning approach for surface-mount device production. In G. Nicosia, P. Pardalos, G. Giuffrida, R. Umeton, & V. Sciacca (Eds.), *Machine learning, optimization, and data science* (Vol. 11331, pp. 254–263). Springer International Publishing. doi: 10.1007/978-3-030-13709-0_21
- Jourdan, N., Longard, L., Biegel, T., & Metternich, J. (2021). *Machine learning for intelligent maintenance and quality control: A review of existing datasets and corresponding use cases*. Institutionelles Repository der Leibniz Universität Hannover. doi: 10.15488/11280
- Liukkonen, M., Havia, E., & Hiltunen, Y. (2012). Computational intelligence in mass soldering of electronics – a survey. *Expert Systems with Applications*, 39(10), 9928-9937. doi: <https://doi.org/10.1016/j.eswa.2012.02.100>
- Mayr, A., Kißkalt, D., Meiners, M., Lutz, B., Schäfer, F., Seidel, R., ... Franke, J. (2019). Machine learning in production – potentials, challenges and exemplary applications. *Procedia CIRP*, 86, 49–54. doi: 10.1016/j.procir.2020.01.035
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85), 2825–2830.
- PHM Society. (2022). *Data challenge: 7th european conference of the prognostics and health management society 2022*. Retrieved 18.06.2022, from <https://phm-europe.org/data-challenge>
- Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48, 144-156. (Special Issue on Smart Manufacturing) doi: <https://doi.org/10.1016/j.jmsy.2018.01.003>
- Zakaria, S., Amir, A., Yaakob, N., & Nazemi, S. (2020). Automated detection of printed circuit boards (pcb) defects by using machine learning in electronic manufacturing: Current approaches. In *Iop conference series: Materials science and engineering* (Vol. 767, p. 012064).

A Novel Methodology for Health Assessment in Printed Circuit Boards

John Taco¹, Prayag Gore¹, Takanobu Minami¹, Pradeep Kundu¹, Alexander Suer¹, Jay Lee¹

¹*Center for Intelligent Maintenance Systems, Department of Mechanical and Materials Engineering, University of Cincinnati, Ohio, USA*

tacolojo@mail.uc.edu, gorepa@mail.uc.edu, minamitu@mail.uc.edu, kundupp@ucmail.uc.edu, suerad@mail.uc.edu, lj2@ucmail.uc.edu

ABSTRACT

The demand for Printed circuit boards (PCBs) has increased due to the rapid change in technology in recent years. Consequently, PCBs health assessment and fault detection play an important role in improving productivity. This study proposed a novel method which focused on feature engineering for health assessment in PCBs. The performance of the proposed method has been validated using data obtained from PHM Europe 2022 data challenge. In this data challenge, PCBs health assessment needs to be performed with data from the Solder Paste Inspection (SPI) and the Automated Optical Inspection (AOI) machine. The challenge has three tasks: 1) Predict the labels of the AOI machine using the SPI data. 2) Using both the SPI and AOI machine data, predict the operator's verification that the AOI machine correctly detected a defect. 3) With the SPI and AOI data, predict the classification of the defective PCBs as either repairable or unrepairable. The component level features are extracted from the original SPI and AOI data which contain the pin level features to solve these tasks. Two machine learning-based classification models, i.e., Light Gradient Boosting Machine (LightGBM) and eXtreme Gradient Boosting (XGBoost), have been used for classification purposes. Training data given by the organizer was divided into 70% training and 30% validation. Based on the validation data, the highest F1-score was observed with LightGBM in Tasks 1 and 2, whereas, in Task 3, the highest F1-score was observed with the XGBoost model. Hence, the LightGBM model has been used in Tasks 1 and 2, and the XGBoost model was developed for Task 3.

Keywords: Diagnosis, PCB, Classification, Feature Engineering

1. INTRODUCTION

A printed circuit board (PCB) goes through the printing machine, which laser prints serial numbers onto the PCB and

applies solder paste according to a predefined structure. The PCB production line is equipped with automated, integrated and fully connected machines that gather data at different stages of production. The electronic components of an electric circuit board (ECB) rely on the solder joint to provide the electrical connection to the PCB (Lee et al., 2002). PCBs demand has been increased due to digitalization and the implementation of Industry 4.0. Thus, their reliability needs to be improved to increase productivity. Consequently, PCBs fault diagnosis plays an important role in technological development. Several approaches have been developed for PCBs health assessment and fault diagnosis in the past few years.

For instance, Wu et al. (2021) proposed two target detection network approaches for health assessment and fault detection of PCBs. Image datasets of PCBs with 6 kinds of defects are used for training and validation purposes. The proposed methodologies show high prediction performance in both health assessment and fault detection tasks. Nayak et al. (2017) suggests a PCB fault detection algorithm using image processing. PCB images are used to train the algorithm and detect the faults before the etching process. Al-Obaidy et al. (2017) developed a fault detection model for PCBs employing thermal image processing. Three algorithms were performed: multilayer perceptron, adaptive neuron-fuzzy and support vector machine. In Chang et al.'s work (2019), Solder Paste Inspection (SPI) data is used to enhance the solder joint's detection performance, which is a type of defect for PCB. This study indicates that the combination of SPI and Automated Inspection (AOI) can make a system with high detectability for PCB faults.

Our present work proposes a novel technique for PCB health assessment using machine learning classifiers such as LightGBM and XGBoost. The significant contribution of this study is to identify novel features for an accurate health assessment. The data from the PHM Europe (PHME) 2022 data challenge has been used to show the performance of the proposed methodology. The PHME data challenge for the year 2022 features a dataset from an actual industrial

John Taco et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

application of an ECB production system (PHM Europe Data Challenge, 2022), as shown in Figure 1. The PCB is transferred to a SPI machine, which assesses the quality of the solder by determining various characteristics of the solder's placement, such as the volume, area, size, and offset from the desired position of solder. It extracts rich metadata up to the pin level of each component in every image of a particular panel. The data is indexed according to the laser inscriptions on the PCB. The PCB now goes through the Surface Mount Device (SMD) placement machine which assembles various components on the PCB's wet solder paste at predefined locations. This assembled PCB goes through a reflow oven which reflows the solder paste to create permanent solder joints between the PCB and the assembled components. Figure 2 illustrates a PCB after solder and component placement (PHM Europe Data Challenge, 2022).

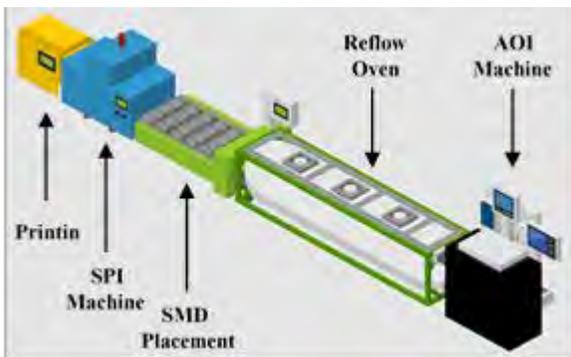


Figure 1: ECB Manufacturing Process (PHM Europe Data Challenge 2022)

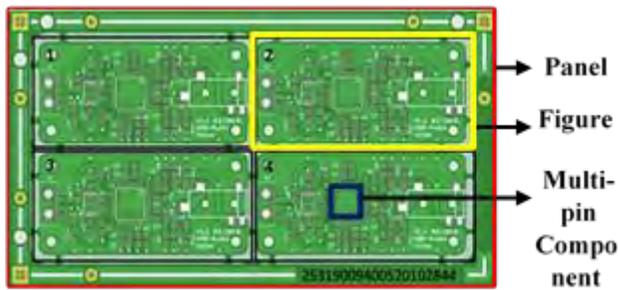


Figure 2: Electronic Circuit Board Panel Process (PHM Europe Data Challenge 2022)

Once the components form permanent solder joints, ECBs go through an AOI machine. The AOI machine automates the visual solder joint inspection and therefore requires the extraction of information from the solder joint surface (Kim et al., 1996). This machine uses a non-contact visual inspection method to detect and classify a solder joint's surface defects (Moganti et al., 1996). It inspects different aspects of the PCB after component placement and solder reflow, like, misalignment, size and fillets of solders, missing components or solder paste, etc.

After this stage of AOI inspection, operators (humans) are employed to verify that the AOI machine did not falsely label the PCB as defective. The operators also further classify the truly defective PCBs into different types before considering any repair work. The structure of this inspection process is outlined in Figure 3. Data is provided with labeled data from SPI and AOI machines. Using this data, three tasks have to be completed 1) Using only the SPI data, predict the labels of the AOI machine. 2) With both the SPI and AOI machine data, predict whether the operator will verify that the AOI machine correctly detected a defect or will label it as a false positive. 3) Classify, with the SPI and AOI data, the defective PCBs as either non-repairable or as PCBs that should not be scrapped.

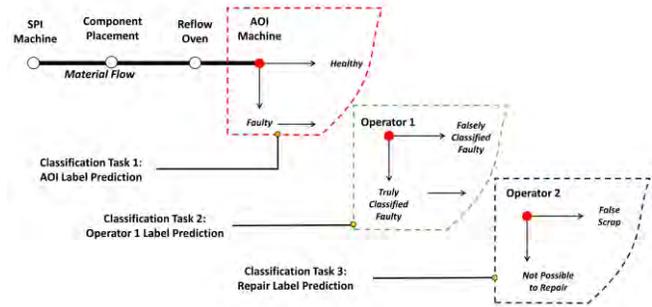


Figure 3: Data analysis for replacing actual inspection tasks

The dataset consists of two data types, each from a different source: the SPI machine and the AOI machine. The data from the SPI contains the attributes of the solder paste placed on the PCBs. The AOI data contains the AOI labels, operator labels, and repair labels. Every dataset contains a panel I.D., a figure I.D., and a component I.D. These three I.D.s can be used together as unique I.D.s for indexing the data. Any classification task would predict labels for the combination of these three I.D.s. This is helpful for predicting classification labels at the component level. Any unique I.D. from the SPI dataset, which can also be found in the AOI dataset, is labelled as faulty. If the unique I.D. from the SPI dataset is not found in the AOI dataset, the respective component is considered healthy.

2. PROPOSED METHODOLOGY

Figure 4 shows the proposed methodology for solving the three tasks in the present study. The data is first cleaned for any instances of missing values. It is then indexed according to the combination of the Panel_ID, Figure_ID, and Component_ID. All data columns are then converted to numeric formats, and the ones which cannot are discarded.

The proposed methodology focuses mostly on feature engineering and uses readily available machine learning libraries for model building. Individual task features have been engineered and ranked according to their suitability for performing the given tasks. Feature engineering combines

multiple raw data features by applying various mathematical operations.

Indexing the data in the aforementioned manner enables data analysis at the component level. The original dataset, however, contains observations at the pin level. Hence, each combination of the aforementioned I.D.s will have the number of observations equaling the number of pins for a particular component. To combine the data of all pins of a single component, aggregation of each raw data variable is performed. This aggregation leads to the extraction of statistical features like mean, standard deviation, variance, etc.

Feature engineering was performed over the raw data by extracting the statistical features by aggregating the data at the component level. Finally, this dataset was then divided into a training data set with 70% of the whole data and a validation dataset with the remaining 30% while keeping the ratio of healthy to faulty classes equal to that of the original dataset. This preservation of the ratio is done by stratifying the class labels.

The proposed methodology uses two tree-based gradient boosting algorithms, LightGBM and XGBoost. After an F1-score comparison of these algorithms for different tasks, the LightGBM model was chosen and used for classification tasks 1 and 2. For task 3, the XGBoost classifier model was used. Figure 4 outlines the proposed methodology for model training and evaluation.

A detailed description of the features and the reasoning for choosing different classifier models for different tasks is discussed in the next section.

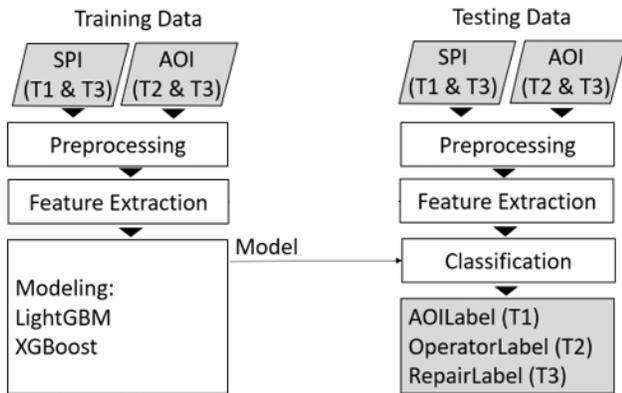


Figure 4: Proposed Methodology

3. RESULTS AND DISCUSSION

3.1. Task 1: Predicting AOI Labels

The classifier model for Task 1 is expected to predict whether a data instance similar to the ones in the SPI raw dataset would be classified by the AOI machine as healthy or faulty.

Special emphasis is given to position-based features since misalignment of solder paste is a leading factor in the PCB being classified as faulty (Chuang et al., 2010)

3.1.1. Data preprocessing:

The data is first cleaned by deleting all instances with null values in Panel_ID, Figure_ID and Component_ID. Data rows containing null values are also erased from the dataset. All data columns are then converted to numeric format to perform arithmetic operations.

3.1.2. Feature Extraction:

This step would include variable generation and statistical feature extraction.

Variable Generation:

Spatial and positional variables have been generated using arithmetic combinations of raw data variables. Along with 12 raw data variables and 6 additional variables, namely, the Total Height, Hypotenuse, Polar Coordinate, Circular Area, Rectangular area and Offset Area, are generated. These variables are generated at the pin level, as shown in the variable column of Table 1.

Statistical Features:

The raw data variables are aggregated with the generated variables at the component level. Several rows of observations for different pins of a single component are aggregated to extract statistical values. The statistical features extracted for this task contain: mean, standard deviation, variance, count, minimum value, maximum value and median. This process reduces the size of data while preserving the information from the raw data in terms of statistical values. Table 1 lists all the features extracted from the raw dataset along with the generated data variables.

Table 1: List of Features for Task -1

Pin Level Features			Component Level Features	
Sr. No.	Variable Name	Operation Required	Feature No.	Feature Name
[1]	Volume(%)	NA	01 - 07	{mean, std, var, count, min, max, median}
[2]	Height(um)	NA	08 - 14	{mean, std, var, count, min, max, median}
[3]	Area(%)	NA	15 - 21	{mean, std, var, count, min, max, median}
[4]	OffsetX(%)	NA	22 - 28	{mean, std, var, count, min, max, median}
[5]	OffsetY(%)	NA	29 - 35	{mean, std, var, count, min, max, median}
[6]	SizeX	NA	36 - 42	{mean, std, var, count, min, max, median}
[7]	SizeY	NA	43 - 49	{mean, std, var, count, min, max, median}
[8]	Volume(um ³)	NA	50 - 56	{mean, std, var, count, min, max, median}
[9]	Area(um ²)	NA	57 - 63	{mean, std, var, count, min, max, median}
[10]	Shape(um)	NA	64 - 70	{mean, std, var, count, min, max, median}
[11]	PosX(mm)	NA	71 - 77	{mean, std, var, count, min, max, median}
[12]	PosY(mm)	NA	78 - 84	{mean, std, var, count, min, max, median}
[13]	Hypotenuse	= sqrt((6) ² + (7) ²)	86 - 91	{mean, std, var, count, min, max, median}
[14]	Rectangular Area	= (6) * (7)	92 - 98	{mean, std, var, count, min, max, median}
[15]	Total Height	= (2) + (10)	99 - 105	{mean, std, var, count, min, max, median}
[16]	Circular Area	= π * ((13) ²)	106 - 112	{mean, std, var, count, min, max, median}
[17]	Offset Area	= π * ((13) ² + (5) ²)	113 - 119	{mean, std, var, count, min, max, median}
[18]	Polar Coordinate	= sqrt((11) ² + (12) ²)	120 - 126	{mean, std, var, count, min, max, median}

3.1.3. Modelling:

After statistical feature extraction, the resulting feature pool consists of 126 distinct data features.

The dataset used for training the task 1 classifier is the highly imbalanced SPI data (98.7% Healthy Class, 1.3% Faulty Class). Due to the highly imbalanced nature of SPI data, the simplest classifier would yield high accuracy but low recall models. Hence, a better metric to assess the performance of the classifier would be the F1 score for the minority class (faulty PCBs). The performance of the LightGBM model in terms of F1-score was compared and found to be better than the XGBoost model for task 1, as shown in Table 1.

Table 2: F1-score comparison for Task 1

Type of Classifier	F1-Score	
	Training Data set	Validation Data set
XG-Boost	0.32	0.31
Light-GBM	0.43	0.42

The Light-GBM model for Task 1 considers the values of the hyperparameters as follows: learning rate of 0.1 and boosted trees of 100. Figure 5 represents the top ten features obtained from the LightGBM classifier model while training.

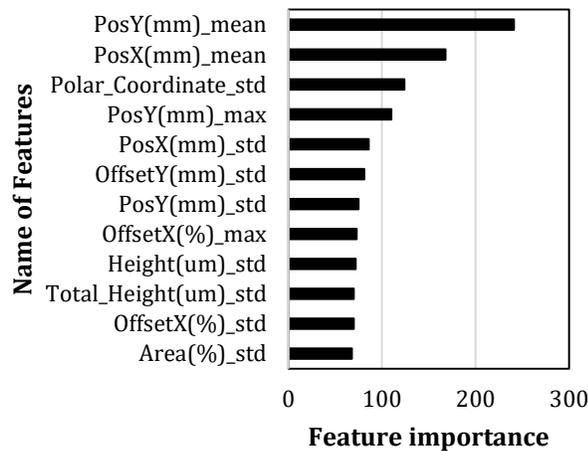


Figure 5: Feature Importance for Task 1

Predictions:

Based on the best trained LightGBM model, an F1-score of 0.44 was observed on the unseen test data set.

3.2. Task 2: Predicting Operator Labels

For task 2, predicting the operator label based on SPI and AOI data is the objective. The presented approach includes steps such as data preprocessing, feature extraction, and modeling.

3.2.1. Data preprocessing:

All samples that have null values in the Panel_ID, Figure_ID or component I.D. are erased. Furthermore, all continuous values that are in a string format are converted into a numeric format.

3.2.2. Feature Extraction:

The feature extraction step considers two datasets: AOI and SPI data. A data pivoting technique (Kim et al., 2019) is applied to the AOI data. Eleven features corresponding to the eleven AOI fault modes (AOILabel) in the training data are generated by this technique. The value of the new features is the count of each failure mode per component. In Figure 6, an example of data pivoting for two fault modes is shown.

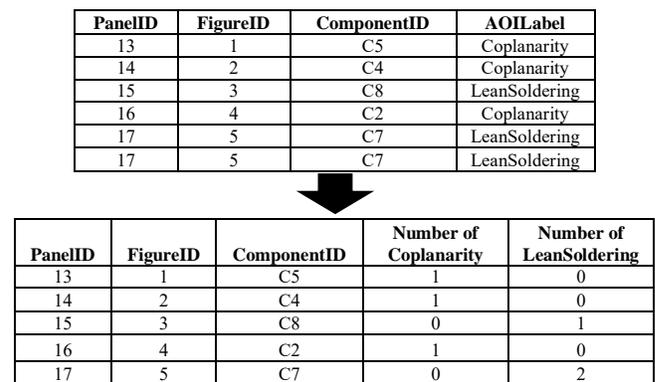


Figure 6 Data conversion to pivot table

Moreover, the total number of AOI fault modes and the number of unique AOI fault modes per component are also considered features. Additionally, the component type and number are extracted from the component I.D. variable. This feature can be extracted either from the SPI or AOI data. The list of extracted features is described in Table 3.

Table 3 Features list for AOI Data

Feature Number	Feature Name
1	Number of Soldered
2	Number of UnSoldered
3	Number of Coplanarity
4	Number of LeanSoldering
5	Number of Translated
6	Number of Size
7	Number of Misaligned
8	Number of Missing
9	Number of Broken
10	Number of Jumper
11	Number of Polarity
12	Total number of AOI labels
13	Total number of unique AOI labels
14	Component type (ex. C, R)
15	Component number (ex. 5)

Due to the low frequency in the training data of the AOI fault modes "Missing", "Broken", "Jumper", and "Polarity", they are considered as "others" and grouped together. This grouping approach reduces the number of features generated from AOI from fifteen to twelve.

For SPI data, statistical features are extracted from the pin level to the component level. Using SPI data, features such as minimum, maximum, and mean values of Volume(%), Height(um), Area(%), OffsetX(%), OffsetY(%), SizeX, SizeY, and Shape(um) are extracted.

Table 4 Features list for SPI Data

Feature Number	Feature Name
1-3	Volume(%) (max, min, mean)
4-6	Height(um) (max, min, mean)
7-9	Area(%) (max, min, mean)
10-12	OffsetX(%) (max, min, mean)
13-15	OffsetY(%) (max, min, mean)
16-18	SizeX (max, min, mean)
19-21	SizeY (max, min, mean)
22-24	Shape(um) (max, min, mean)

3.2.3. Modeling

Similar to task 1, two algorithms are used and compared for modelling: LightGBM and XGBoost. Each algorithm is trained using cross-validation ensemble to enhance robustness.

Furthermore, three combinations of features were performed: only AOI features, only SPI features, and both AOI and SPI features. As shown in Table 5, models that use only the AOI features perform better than the other attempted approaches.

The model leads to overfitting using AOI-SPI features and only using SPI features. Thus, only the AOI features are used, and LightGBM is selected due to its higher F1 performance than XGBoost.

Table 5 F1-score calculation using different classifiers

Type of Classifier	AOI data		SPI data		AOI-SPI data	
	Train	Val	Train	Val	Train	Val
XGBoost F1-Score	0.66	0.68	0.65	0.34	0.78	0.62
LightGBM F1-Score	0.69	0.69	0.66	0.33	0.71	0.61

The LightGBM model for Task 2 considers the hyperparameters such as a learning rate of 0.2 and a count of boosted trees of 5000. The feature importance ranking given by the LightGBM classification model is shown in Figure 7.

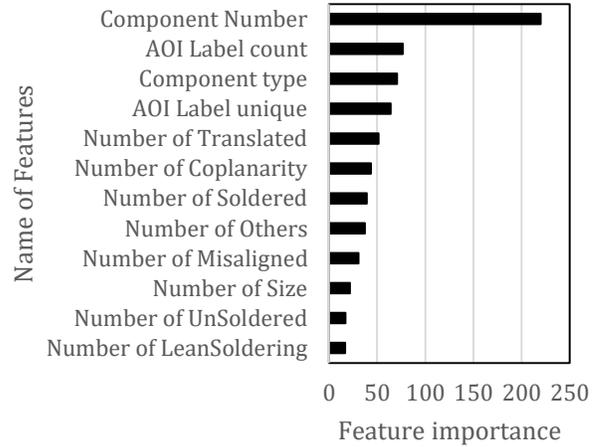


Figure 7 Feature importance ranking

Predictions:

The classification model gives probabilities as its output. Then, the probabilities are used to calculate the optimal threshold to maximize the F1-score of the validation data. The thresholds are obtained using the package 'metric' from the Sklearn library. The optimal threshold is obtained by iterating all possible thresholds and selecting the one that provides the maximum F1-score for the validation data. The approach for task 2 gives an F1-score of 0.48 in the unseen test data.

3.3. Task 3: Predicting Repair Labels

The objective of Task 3 is to predict the Repair Labels based on SPI and AOI data. Similar to other tasks, the approach includes steps such as data preprocessing, feature extraction, and modeling.

3.3.1. Data preprocessing:

Similar to task 2, null values in the Panel_ID, Figure_ID or Component_ID are erased from SPI data. Also, all continuous values have been converted into the float type.

3.3.2. Feature extraction

The solution for task 3 first uses the component level features instead of the pin level features like the solutions for the other tasks. In this task, 17 variables are already available from SPI and AOI data, and additional 11 variables shown as serial numbers 18 to 28 in Table 6 are formed using the existing 17 variables. From these 28 pin level variables, component level features have been created from pin level features which have the same Panel_ID, Figure_ID, and Component_ID. These features are calculated based on statistical metrics such as mean, sum, standard deviation, maximum, minimum, peak-to-peak, median, and count. A total of 153 component-level features have been extracted, as shown in Table 6.

Table 6 Features extracted for Task 3

Pin Level Features				Component Level Features		
Sr. No.	Variable Name	SPI	AOI	Operation Required	Feature No.	Feature Name
[1]	Volume(%)	X			01-06	{mean, std, var, count, min, max, median}
[2]	Height(um)	X			07-12	{mean, std, var, count, min, max, median}
[3]	Area(%)	X			13-18	{mean, std, var, count, min, max, median}
[4]	OffsetX(%)	X			19-24	{mean, std, var, count, min, max, median}
[5]	OffsetY(%)	X			25-30	{mean, std, var, count, min, max, median}
[6]	SizeX	X			31-36	{mean, std, var, count, min, max, median}
[7]	SizeY	X			37-42	{mean, std, var, count, min, max, median}
[8]	Volume(um3)	X			43-48	{mean, std, var, count, min, max, median}
[9]	Area(um2)	X			49-54	{mean, std, var, count, min, max, median}
[10]	Shape(um)	X			55-60	{mean, std, var, count, min, max, median}
[11]	PosX(mm)	X			61-66	{mean, std, var, count, min, max, median}
[12]	PosY(mm)	X			67-72	{mean, std, var, count, min, max, median}
[13]	PinNumber	X			73-74	{median, count}
[14]	PadType_*		X		75-76	{count} (* = 0, 10)
[15]	PinNumber_*		X		77-80	{count} (* = 0, 1, 2, 3)
[16]	AOILabel_*		X		81-86	{count} (* = 1, 2, 3, 4, 5, 6)
[17]	AOILabel	X			87	{count}
[18]	OffsetX and Y	X		= sqrt([4]^2 + [5]^2)	88-93	{mean, std, var, count, min, max, median}
[19]	SizeX and Y	X		= sqrt([6]^2 + [7]^2)	94-99	{mean, std, var, count, min, max, median}
[20]	Volume / area	X		= [8] / [9]	100-105	{mean, std, var, count, min, max, median}
[21]	Volume / Volume	X		= [9] / [1]	106-111	{mean, std, var, count, min, max, median}
[22]	PosX and Y	X		= sqrt([11]^2 + [12]^2)	112-117	{mean, std, var, count, min, max, median}
[23]	OffsetX/OffsetY	X		= [4]/[5]	118-123	{mean, std, var, count, min, max, median}
[24]	OffsetY/SizeX	X		= [4]/[6]	124-129	{mean, std, var, count, min, max, median}
[25]	OffsetY/SizeY	X		= [5]/[7]	130-135	{mean, std, var, count, min, max, median}
[26]	Volume/OffsetX	X		= [11]/[4]	136-141	{mean, std, var, count, min, max, median}
[27]	SizeY/PosY	X		= [7]/[12]	142-147	{mean, std, var, count, min, max, median}
[28]	PosX/PosY	X		= [11]/[12]	148-153	{mean, std, var, count, min, max, median}

3.3.3. Modeling

Two classification algorithms were tested, XGBoost and LightGBM, and the best performing model was selected. Based on these 153 features, Table 7 shows the F1-score obtained using each model. Based on the F1-score, XGBoost was found to perform better and hence has been used for model development.

Table 7 F1-score calculation using different classifiers

Type of Classifier	F1-Score	
	Training	Validation
XGBoost	0.99	0.90
LightGBM	0.87	0.89

The XGBoost model for Task 3 considers the hyperparameters such as a learning rate of 0.1 and a maximum of boosted trees of 100. The feature importance ranking given by XGBoost for this task is shown in Figure 8.

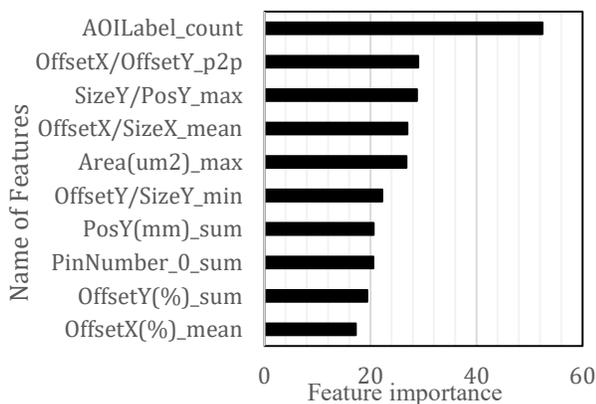


Figure 8 Feature importance ranking

Predictions:

The approach for task 3 gives an F1-score of 0.78 in the unseen test data.

4. CONCLUSIONS

This work has developed methodologies to detect printed circuit board manufacturing defects based on SPI and AOI data released by the PHM Society (2022). The component level features are extracted from original SPI and AOI data which contain the pin level features. The PCB health assessment problem has been divided into three tasks, and based on the extracted features, two machine learning algorithms, LightGBM and XGBoost, have been applied. It was determined that the models using LightGBM for tasks 1 and 2 had a better F1-score on the validation data set than the XGBoost models. Hence, LightGBM classification models were selected for first two tasks. The XGBoost model has been used in task 3, due to its higher F1-score compared to the LightGBM model's results. The F1-score obtained from Task 1, Task 2 and Task 3 are 0.44, 0.48 and 0.78 respectively.

REFERENCES

Al-Obaidy, F., Yazdani, F., and Mohammadi, F. A., (2017). Fault detection using thermal image based on soft computing methods: Comparative study. *Microelectronics Reliability*, vol. 71, pp. 56-64, doi: <https://doi.org/10.1016/j.microrel.2017.02.013>.

Chang, Y. M., Wei, C. C., Chen, J., & Hsieh, P. (2019). An implementation of health prediction in SMT solder joint via machine learning. *IEEE international conference on big data and smart computing*, pp. 1-4.

Chuang, S. F., Chang, W. T., Lin, C. C., & Tarng, Y. S. (2010). Misalignment inspection of multilayer PCBs with an automated X-ray machine vision system. *The International Journal of Advanced Manufacturing Technology*, vol. 51(9), pp. 995-1008.

Kim, J. H., Cho, H. S., and Kim, S. (1996). Pattern Classification of Solder Joint Images Using a Correlation Neural Network. *Engineering Applications of Artificial Intelligence*, vol. 9(6), pp. 655-669, doi: [https://doi.org/10.1016/S0952-1976\(96\)00046-2](https://doi.org/10.1016/S0952-1976(96)00046-2).

Lee, Z., and Lo, R. (2002). Application of vision image cooperated with multi-light sources to recognition of solder joints of PCB.

Moganti, M., Ercal, F., Dagli, C. H., and Tsunekawa, S. (1996). Automatic PCB Inspection Algorithms: A Survey. *Computer Vision and Image Understanding*, vol. 63, no. 2, pp. 287-313. doi: <https://doi.org/10.1006/cviu.1996.0020>.

Nayak, J. P. R., Anitha, K., Parameshachari, B. D., Banu, R., and Rashmi, P. (2017). PCB Fault Detection Using Image Processing. *Materials Science and Engineering Conference Series*, vol. 225, no. 1, pp. 012244. doi: 10.1088/1757-899X/225/1/012244.

PHM Europe Data Challenge. (2022). Data Challenge Website. *7th European Conference of The Prognostics and Health Management Society*. [Online], Available: <https://phm-europe.org/data-challenge>

Wu, X., Ge, Y., Zhang, Q., and Zhang, D. (2021). PCB Defect Detection Using Deep Learning Methods, *IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 873-876, doi: 10.1109/CSCWD49262.2021.9437846.

BIOGRAPHIES



John Taco received his B.S. degree in mechanical engineering from Pontifical Catholic University of Peru, Lima, Peru, in 2018. He is currently pursuing his M.S. and PhD degrees in mechanical engineering with the University of

Cincinnati, Cincinnati, OH, USA. His research interests include deep learning, prognostics, health management, and industrial A.I.



Prayag Gore received his B.S. degree in Mechanical Engineering from University of Pune, India in 2018. He is currently pursuing his M.S. and PhD degrees in Mechanical Engineering at the University of Cincinnati, OH, USA. His research

areas and research interests include machine learning, prognostics and health management, condition monitoring, machine vision and industrial A.I.



Takanobu Minami received his B.S. and M.S. degree in mechanical engineering from Kyoto University in 2008 and in 2011, respectively. Currently, he is pursuing his Ph.D. degree in mechanical engineering with the University of

Cincinnati, Cincinnati, OH, USA and employed as engineer in Komatsu Ltd. His research interests include machine learning, deep learning, prognostics and health management, and industrial A.I.



Pradeep Kundu is a Postdoctoral Fellow at the IMS Center, University of Cincinnati. Before joining IMS, he worked as a research associate from Dec. 2020 to Jan. 2022 on an EPSRC-funded project titled "A Multiscale Digital Twin-Driven

Smart Manufacturing System for High Value-Added Products" at the University of Strathclyde, U.K. Pradeep completed his PhD study in 2020 in the specialization of Prognosis and Health Management from the Department of Mechanical Engineering at the Indian Institute of Technology (IIT) Delhi, India. In 2019, he received a Visiting Research Fellow grant from SERB, Govt. of India, to carry out his PhD research work at the University of Alberta, Canada. Pradeep has published more than 20 articles in reputed academic journals and conferences such as Mechanical Systems and Signal Processing, Journal of Intelligent Manufacturing, Structural Health Monitoring, etc. His research interests are Industrial Artificial Intelligence, Cyber-Physical Systems, Digital Twins and Smart Manufacturing, Fault Diagnosis and Prognosis and Reliability Engineering,



Alexander Suer received his B.S. in mechanical engineering from the University of Cincinnati, Cincinnati, OH, USA in 2022. He is pursuing his M.S. and Ph.D. degree in mechanical engineering with the University of Cincinnati. His research interests include machine learning, prognostics and health management, industrial A.I., and robotics.



Jay Lee received the B.S. degree in electrical engineering from Tamkang University, Taipei City, Taiwan, and the M.S. degree in electrical engineering from the University of Wisconsin–Madison, Madison, WI, USA, and the Ph.D. degree from George Washington University,

Washington, DC, USA. He is currently an Ohio Eminent Scholar, a L. W. Scott Alter Chair Professor, and a Distinguished University Professor with the University of Cincinnati, Cincinnati, OH, USA. He is the Founding Director of the National Science Foundation (NSF) Industry/University Cooperative Research Center on Intelligent Maintenance Systems, which is a multicampus NSF Industry/University Cooperative Research Center consisting of the University of Cincinnati (lead institution), the University of Michigan, Ann Arbor, MI, USA, the Missouri University of S&T, Rolla, MO, USA, and the University of Texas-Austin, Austin, TX, USA. Since its inception in 2001, the Center has been supported by more than 90 global companies and was ranked with the highest economic impact (270:1) by NSF Economics Impacts Report in 2012.

Prediction of Production Line Status for Printed Circuit Boards

Haichuan Tang¹, Yin Tian², Junyan Dai³, Yuan Wang⁴, Jianli Cong⁵, Qi Liu⁶, Xuejun Zhao⁷, Yunxiao Fu⁸

^{1,2,6,7,8} *CRRC Academy, Beijing, 100161, China*

thc@crrc.tech

ty@crrc.tech

lq@crrc.tech

zsj@crrc.tech

fyx@crrc.tech

³*Rutgers University, New Brunswick, NJ, 08854, USA*

jd1394@scarletmail.rutgers.edu

⁴*Southern University of Science and Technology, Shenzhen, Guangdong, 518000, China*

wang.skoud@gmail.com

⁵*Southwest Jiaotong University, Chengdu, Sichuan, 610031, China*

jlcong2019@my.swjtu.edu.cn

ABSTRACT

This paper focuses on the problem of predicting production line status for Printed Circuit Boards (PCBs). The problem contains three prediction tasks regarding PCB producing process. Firstly, data exploration is carried out and it reveals several data challenges, including data imbalance, data noise, small sample size, and component difference. To predict production line status for components of PCBs using records of inspection on pins, we proposed two possible feature extraction methods to compress the pin-level data into component-level. A statistical feature extraction method, which retrieves descriptive statistics such as mean, standard deviation, maximum, and minimum of pins on the same component, is applied to Task 1, while a PinNumber-based feature extraction method, which keep original values for 2-pin components, is applied to Task3. In addition, a neural-net model with feeding imbalance control is established for Task 1. and a random forests model is applied for both Task 2 and Task 3. Moreover, a threshold moving technique is proposed to optimize the threshold selection. Finally, the result shows that our models achieved f1-scores of 0.44, 0.54, and 0.71 on the test set for the three tasks, respectively.

1. INTRODUCTION

The PCB is a platform installed with semiconductor chips, capacitors, and other components, providing electrical

interconnection between components. It is used in virtually all electronic products. Accurate prediction of PCB production line status can effectively reduce the production cost. During the production line, the PCB goes through the printing machine, solder paste inspection (SPI), surface mount device placement, reflow oven, and automatic optical inspection (AOI) in sequence. The production line contains two inspections: the SPI and the AOI. The SPI measures the shape of the solder paste and detects possible problems with the solder paste after it is placed by the printing machine. The AOI checks defects that might occur after the PCB goes through surface mount device placement and reflow oven. The AOI produces three labels as follows:

- 1) AOI label: indicating the type of defects detected by the AOI machine.
- 2) Operator label: assigned by the human operator, indicating whether the AOI machine raised a false defect or not.
- 3) Repair label: assigned by the repairment operator, indicating the repair action.

There are many benefits for predicting the above-mentioned production status. For example, by accurately predicting operator label and repair label, operators can improve their efficiency by arranging their work orders based on the prediction results. The problem of predicting the above three labels using SPI and AOI data is posed in PHME 2022 Data Challenge (PHME Data Challenge, 2022).

Haichuan Tang et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Predicting production line status is a challenging problem due to its extremely imbalanced dataset, where the population of the normal status data is far more than that of the faulty data. As imbalanced data is common in real world problems, many methods have been proposed by prior studies to deal with it. Data resampling is often considered as the first choice to solve the data imbalance problems (Batista, et al., 2004). The data resampling methods include randomly oversampling the minority class, randomly undersampling the majority class, and other advanced approaches such as the synthetic minority oversampling technique (Chawla, et al., 2002) and adaptive synthetic sampling (He, et al., 2008). However, resampling the data may lead to a worse model for some reasons: 1) undersampling may discard useful information and 2) oversampling may increase the chance of overfitting and the learning time. Another approach to deal with imbalanced data is moving decision thresholds of a learned model. This approach has been implemented and tested as a better alternative for resampling (Provost, 2000; Maloof, 2003).

The problem of PHME 2022 Data Challenge is even more challenging in that the proposed models need to consider not only the high imbalance ratio, and noise in the dataset, but also the different numbers of pins for different components, which means specific feature extraction is needed to unify the feature structure.

In our solution, we propose a machine learning-based method to predict production line status using data from previous production steps. A statistical feature extraction and a PinNumber-based feature extraction method are proposed to reconstruct pin-level data into component-level data to perform predictions on PCB components. The statistical feature extraction method retrieves descriptive statistics such as mean, standard deviation, maximum, and minimum values of pins on the same component. The PinNumber-based feature extraction method treats PCB components differently based on the number of pins they contain. This is inspired by the data exploration that most of the PCB components have 2 pins. To deal with the imbalanced dataset, we introduce a neural network model with feeding imbalance control and a random forests method with a threshold moving technique. As a result, our proposed method achieved f1-scores of 0.44, 0.54, and 0.71 on the test for the three tasks posed by this data challenge.

The rest of the paper is organized as follows: the problem definition, datasets, and scoring functions are described in Section 2. Data exploration and preprocessing methods are provided in Section 3. In Section 4, classification methods are introduced. Results are discussed in Section 5 and conclusions are drawn in Section 6.

2. DATA CHALLENGE DESCRIPTION

2.1. Problem Definition

The data challenge focuses on predicting labels produced by the AOI. As the AOI produces three labels, the problem is divided into three tasks as follows:

- 1) Task 1: Predict whether any defects in PCB components will be detected by the AOI machine.
- 2) Task 2: Predict whether the AOI machine will raise a false defect according to the human operator.
- 3) Task 3: Predict the repair label assigned by the repairment operator.

Task 1 and Task 2 are binary classification problems, whereas Task 3 is the only multi-class classification problem in this data challenge.

2.2. Datasets

The datasets of this challenge were collected from the SPI and AOI. After removing the data with a null value, the SPI data contains 5,985,381 records on 1,969,523 components, while AOI data contains 31,617 records on 27,514 components. This is because a component may have several pins used for soldering. For a detailed description of the datasets, please refer to <https://phm-europe.org/data-challenge>.

2.3. Scoring

In this challenge, the F1-score is used to evaluate the model performance. The F1-score is calculated based on the ground truth and the predicted labels. The relating functions are listed as follows:

$$F1_l = 2 \cdot \frac{precision_l \cdot recall_l}{precision_l + recall_l} \quad (1)$$

$$precision_l = \frac{TP_l}{TP_l + FP_l} \quad (2)$$

$$recall_l = \frac{TP_l}{TP_l + FN_l} \quad (3)$$

Where TP represents the true positive, FP represents the false positive, FN represents the false negative, and the sub-notation l denotes the positive class. The specific scoring function for each task is listed in Table 1. Task 1 considers the components in the AOI dataset as the positive class. Task 2 considers the “Bad” operator label as the positive class. In Task 3, which is a multi-class classification problem, the average of the F1-scores using “NotPossibleToRepair” and “FalseScrap” as positive classes respectively is calculated. The final score is the average of the F1-scores computed from the three tasks.

Table 1. Scoring function for each task

Task No.	Scoring Function
Task 1	$F1_{inAOI}$
Task 2	$F1_{Bad}$
Task 3	$(F1_{FalseScrap} + F1_{NotPossibleToRepair}) / 2$

3. DATA EXPLORATION AND PREPROCESSING

In this section, we explore the original dataset to find a potential design basis for the data preprocessing methods as well as the prediction models. According to our exploration, we propose two feature extraction methods to reduce the data redundancy and construct a proper data frame to be used as model input.

3.1. Data Exploration

To find the correlation between each two data columns, we plot two-dimensional dot charts using one column for the x-axis and the other for the y-axis. An example of the dot charts is shown in Figure 1. The x-axis represents the value for “Height(um)” in SPI data, while the y-axis denotes the “Volume(um³)” value. In Figure 1, the dots constitute multiple linear patterns, which we assume are caused by different component characteristics. Thus, we group the components into 14 types based on the letters in the ComponentID. For example, ComponentIDs {“BC1”, “BC2”, “BC3”, “BC4”} belong to the component type “BC”. The different colors in Figure 1 indicate the component types, which validates our assumption.

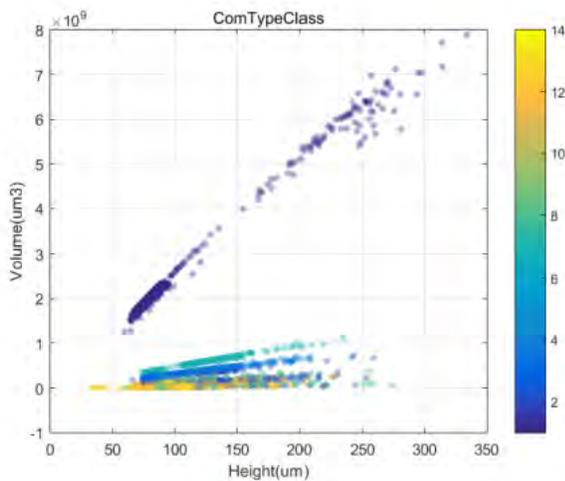


Figure 1. Scatter plots of Volume(um³) and Height(um) with component types in different colors.

Since this challenge focuses on component-level prediction when the original datasets stores pin-level records, we need to convert the original records to component-level features. Figure 2 below shows the frequency distribution of the number of pins in each component in the SPI dataset, where 2-pin components are the majority.

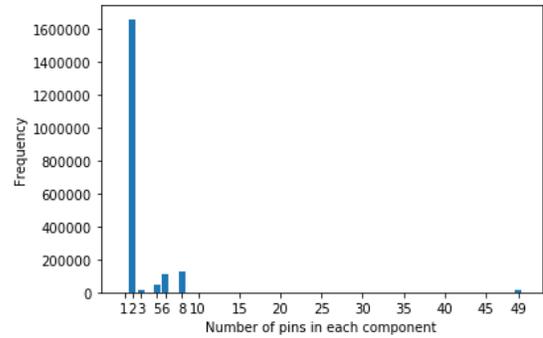


Figure 2. Frequency distribution of the number of pins in each component.

For task 2, the distribution of ‘OperatorLabel’ concerning PosX/Y is presented in Figure 3.

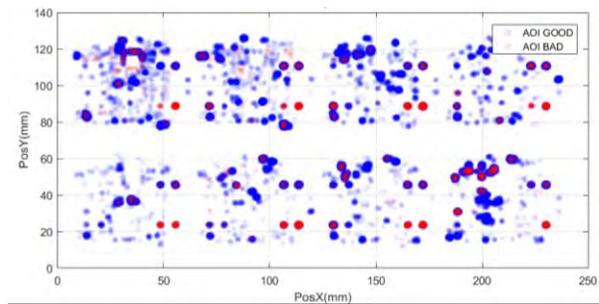


Figure 3: The ‘OperatorLabel’ distribution with respect to PosX/Y.

Some essential features of the dataset are summarized as follows:

- **Highly imbalanced.** The ratio between the majority and minority classes is quite high, 250:1 for Task #1 for example.
- **Quite noisy.** There is a lot of noise inside the dataset.
- **Small sample size.** For Task #2 and #3, the data sample sizes are quite small.
- **Component difference.** There are 14 different types of components and the number of pins of the components varies from 2 to 49. Also, it can be found that different components may have very different feature distributions, and are related to a different defect probability.

3.2. Statistical Feature Extraction

SPI dataset contains measurements regarding welding paste characteristics (i.e., volume, height, area, offset, size) for every pin. To retrieve features for components, we group all measurements of pins in the same component and compute their statistical features (i.e., mean, standard deviation, minimum, maximum). Based on the data exploration above, we also consider the component type and the number of pins as two important features. The pseudo-code for merging pin-

level records into components and the statistical feature extraction algorithm is shown in Algorithm 1.

Algorithm 1: get stat features(*spi*)

```

1  com_ids = get_unique_com_ids(spi)
2  rows = empty list
3  for id in com_ids:
4    panel_id = id[0]
5    figure_id = id[1]
6    com_id = id[2]
7    spi_temp = spi[(spi.PanelID==panel_id) &
(spi.FigureID==figure_id) &
(spi.ComponentID==com_id)]
8    means = get_mean(spi_temp)
9    stds = get_std(spi_temp)
10   max_min = get_max(spi_temp) - get_min(spi_temp)
11   com_type = get_comtype(com_id)
12   com_pin_num = len(spi_temp)
13   rows.append([panel_id, figure_id, com_id,
com_type, com_pin_num, means, stds, max_min])
14 return rows
    
```

3.3. PinNumber-based Feature Extraction

As shown in Figure 2, most components in the SPI dataset contain 2 pins. Thus, we keep all original measurements for 2-pin components to ensure that no information regarding the majority is discarded. For the components with only 1 pin, we duplicate the measurements to form a unified structure. For the rest components with more than 2 pins, we adopt the maximum and minimum values for each measurement as follows:

$$\begin{cases} \hat{X}_{c,f,1} = \min_{p \in [1, P_c]} (X_{c,p,f}) \\ \hat{X}_{c,f,2} = \max_{p \in [1, P_c]} (X_{c,p,f}) \end{cases} \quad \forall c \in \{C\}, \forall f \in \{F\} \quad (3)$$

Where *c* represents a specific component, {*C*} is the complete set for all components during PCB manufacture, *p* is the pin number, *P_c* is the max pin number for a given component *c*, *f* is one of the numerical features, {*F*} is the complete set of numerical features, *X_{c,p,f}* is the original numerical feature with given component and pin number, $\hat{X}_{c,f,1}$ and $\hat{X}_{c,f,2}$ are two new numerical features for a given component to replace original ones.

The pseudo-code for merging pin-level records into components and pin number-based feature extraction algorithm is shown in Algorithm 2.

Algorithm 2: get pinn features(*spi*)

```

1  com_ids = get_unique_com_ids(spi)
2  rows = empty list
3  for id in com_ids:
4    panel_id = id[0]
5    figure_id = id[1]
    
```

```

5  com_id = id[2]
6  spi_temp = spi[(spi.PanelID==panel_id) &
(spi.FigureID==figure_id) &
(spi.ComponentID==com_id)]
7  com_pin_num = len(spi_temp)
8  if com_pin_num == 2:
9    rows.append(spi_temp[0]+spi_temp[1])
10 else if com_pin_num > 2:
11   rows.append(get_max(spi_temp) +
get_min(spi_tem[1]))
12 else:
13   rows.append(spi_temp[0]+spi_temp[0])
14 return rows
    
```

4. CLASSIFICATION METHODS

4.1. Neural Network with Feeding Imbalance Control

The neural-net structure is given in Figure 4. It is a typical forward architecture with multiple dense nets and there is a SoftMax layer before the final output. The key points are given as follows:

- **Categorization.** All continuous features are converted into categorical features according to percentiles division. The percentiles are set as [0, 1, 5, 10:10:90, 95, 99, 100]. This operation will somehow compress the information of some continuous features, such as height(mm) and area(mm). This can be taken as a denoise approach.
- **Binarization.** All categorical features are converted into binary vectors according to one-hot coding and then merged into one binary vector. The label is also encoded by one-hot coding.
- **Training by Feeding Imbalance Control.** According to Section 3.1 Data exploration, we know that the dataset is extremely imbalanced. When training a neural net to fit an imbalanced dataset, we need to control the imbalance ratio of each mini-batch during the training process. As a result, we proposed the concept of Feeding Imbalance Ratio (FIR), which is an implementation of an under-sampling approach.

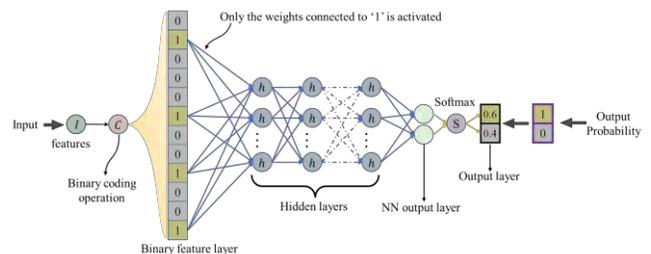


Figure 4. Forward Architecture of Neural-net Model for Probability Prediction

Note that the AOI defect is a rare event, the overall occurrence probability of an AOI defect is about 0.4%. The

ratio between non-event and event is around 250:1. To enhance the performance of the model, we design the training process in a particular way. Firstly, we use the stochastic gradient descent (SGD) method to train the model. Secondly, during the training process, instead of feeding the data randomly, we feed the data with a constraint on the ratio between the majority and minority. We define Feeding Imbalance Ratio as follows:

Definition: Feeding Imbalance Ratio (FIR). The ratio between the majority and minority classes within the resampled mini-batches fed into the NN model during the training process.

Notably, FIR is an important parameter for training the model. If FIR is too large, the dataset fed into the model is imbalanced, and it is hard to learn the feature combination related to the AOI defect. FIR is set to 1 in our case. The training algorithm is given in Table 2.

Table 2. Training Algorithm for probability prediction by feeding imbalance control.

<p>Input:</p> <ul style="list-style-type: none"> ◇ $FIR = 1, batch_size, n_epoch, learning_rate$ ◇ Training dataset: (F, L); ◇ The number of layers and neurons of the neural net; <p>Initialize:</p> <ul style="list-style-type: none"> ◇ Initialize a neural-net $p(* \theta)$; ◇ Split the (F, L) into $(F, L)^+$ and $(F, L)^-$ according to the label, '+' and '-' represent the majority and minority classes, respectively. <p>Main:</p> <p>For i in range (n_epoch), do</p> <p style="padding-left: 20px;">$(F, L)^+ = (F, L)^+.shuffle()$</p> <p style="padding-left: 20px;">$(F, L)^- = (F, L)^-.shuffle()$</p> <p style="padding-left: 40px;">For i in range $(round(size((F, L)^-)/batch_size))$, do</p> <p style="padding-left: 60px;">$(F, L)_i^+ = (F, L)^+.next_batch(batch_size)$</p> <p style="padding-left: 60px;">$(F, L)_i^- = (F, L)^-.next_batch(FIR * batch_size)$</p> <p style="padding-left: 60px;">$F_i^+ = onehot(F_i^+)$</p> <p style="padding-left: 60px;">$L_i^- = onehot(L_i^-)$</p> <p style="padding-left: 60px;">$(F, L)_i = shuffle(concat(F_i^+, L_i^+), concat(F_i^-, L_i^-))$</p> <p style="padding-left: 60px;">Update the parameter θ of $p(* \theta)$ given mini-batch $(F, L)_i$</p> <p style="padding-left: 20px;">End For</p> <p>End For</p> <p>Output: The trained neural-net $p(* \theta)$.</p>
--

4.2. Random Forests

A random forests classifier is an ensemble of tree-structured classifiers. It is an upgraded Bagging algorithm (Breiman, 2001). In Bagging, a bootstrap sample is used to train each weak classifier and the majority vote of the weak classifiers is considered the final prediction. Random Forests further introduces feature randomness to Bagging. Instead of using all features, Random Forests splits each node using a randomly selected subset of features. The Random Forests classifiers are used to solve task 2 and task 3 and are implemented by the scikit-learn python library (Pedregosa et al., 2011).

4.3. Threshold Principles

Since the performance evaluation is based on the F1-score, we have to set a proper threshold for each task once we obtain the continuous output from our model to present our final predicted result set. There are two possible principles for the selection of thresholds:

- **Equal probability.** A threshold is selected assuming that the minority class in the test set has the same occurrence probability as the training set.
- **Best in Training.** A threshold of the test set is selected that performs best in the training set.

Note that the threshold principles work only for Task 1 and Task 2.

5. ANALYSIS OF RESULTS

In this section, we apply our methodologies to PHME 2022 Data Challenge. The scores are displayed in Table 3 below. The train scores are averages of the scores computed through a 5-fold cross-validation and the test scores are the final scores of our team shown on the leaderboard. In addition, a detailed analysis for each task with only the best result of our experiments is demonstrated in this section.

Table 3. F-Scores of the proposed methods

	Task 1	Task 2	Task 3	Final
Training data	0.43	0.68	0.83	0.65
Test data	0.44	0.54	0.71	0.56

5.1. Task 1

Task 1 is focused on predicting whether the AOI machine will raise a defect record based on the SPI information of a component. For the feature extraction part, the main challenge is to unify the pin features of different component types.

In our solution, for each component, regardless of the number of pins, we compress the pin-level features ("*Volume(%)*", "*Height(um)*", "*Area(%)*", "*OffsetX(%)*", "*OffsetY(%)*", "*Volume(um3)*", "*Area(um2)*", "*Shape(um)*", "*PosX(mm)*",

“PosY(mm)”) to component-level by introducing three statistical operators: (1) average, (2) standard deviation and (3) maximal-minimal difference. As a result, the whole SPI dataset is converted into an SPI-Com-level dataset, and each unique component has one data sample with 33 features(the other 3 features are “ComponentID”, “com_type”, and “com_pin_num”).

The Neural-net model is initialized according to the parameters given in Table 4. During the training process, the FIR is fixed to 1.0. The convergence curve is presented in Figure 5 (x-log scale). Note that the initial value of the loss function is 0.5, indicating the initial untrained neural-net outputs randomly. Soon after about 10 training iterations, a quick converging trend can be found, and the converging trend slows down as the learning rate decays.

Table 4. Parameters of Neural-net for Task 1.

	Parameter	Value
1	Number of neurons	256
2	Batch size	32
3	Learning rate	1e-2 to 1e-4
4	F.I.R.	1.0
5	Training epochs	50
6	Train-test split	70% training and 30% test

The F1-scores of the training and test dataset in terms of different thresholds are given in Figure 6. It can be found that the best F1-score is only slightly larger than the test result. More importantly, the related threshold for the best F1-score is almost the same, which is about 0.976. As a result, conclusions can be drawn as follows:

- The best F1-score of the training and test set is about 0.43, and our model is less likely over-fitted. Note that it may be possible to increase the model scale and do more iterations to achieve better performance.

The best threshold to be selected is about 0.976. More generally, the best threshold can be determined according to the one who performs best in the training dataset.

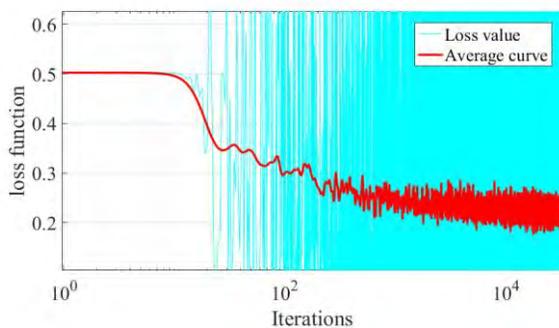


Figure 5: The loss function with respect to training iterations.

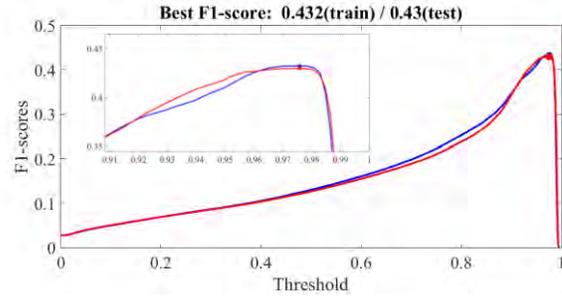


Figure 6: The F1-scores of the training and test dataset.

5.2. Task 2

Task 2 focuses on predicting if the AOI machine will raise a false defect. Thus, we need to remove components that do not exist in the AOI dataset and 27,514 components are left in the training set. Among the 27,514 components, there are 412 components having a “Bad” operator label, and 27,093 components having a “Good” operator label, which is considered highly imbalanced. To tackle the data imbalance issue, we apply a threshold moving technique, best in training, to select the optimized threshold and use it for final testing. Figure 7 validates our threshold moving technique by comparing the predicted test F1-score and the best test F1-score among all possible thresholds. Based on our experiments, a random forests model is built using the selected features: “com_pin_num”, “Result”, “com_type”, “AOILabel”, and “MachineID”. 200 n_estimators, 21 max_depth, and 4 min_samples_split are considered the best hyperparameters for our model using grid-search optimization. Note that one-hot encoding is applied for categorical variables.

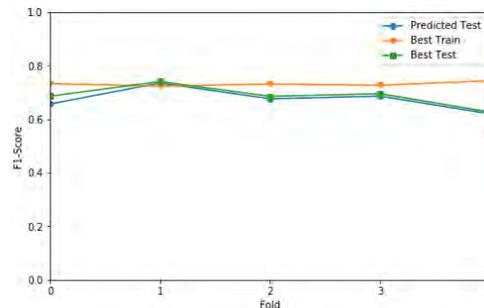


Figure 7. F1-scores for the training set, test set using selected threshold, and the global best f1-score for the test set

5.3. Task 3

Task 3 aims to predict the repair label for only faulty components. Thus, we need to remove components if OperatorLabel is “Good”. There are 412 components left in the training set for task 3. Based on our experiments, a combination of the PinNumber-based feature extraction and random forest algorithm produces the best result. The selected features are “com_pin_num”, “com_type”,

“Volume(%)”, “Height(um)”, “Area(%)”, “OffsetX(%)”, “OffsetY(%)”, “Volume(um3)”, “Area(um2)”, “Shape(um)”, “PosX(mm)”, “PosY(mm)”, “Result”, “AOILabel”, and “MachineID”. In addition, 300 n_estimators and 7 max_depth are considered the best hyperparameters for our model using grid-search optimization.

Using the optimal model parameters, the model performance on the training set of this multi-classification problem has been presented in the form of a confusion matrix, which is shown in Figure 8.

True Label	FalseScrap	121	1	
	NotPossibleToRepair	3	222	
	NotYetClassified	5	4	56
		FalseScrap	NotPossibleToRepair	NotYetClassified
		Predicted Label		

Figure 8. Task 3 confusion matrix using the training set

6. CONCLUSION

This paper focuses on three prediction tasks regarding the PCB manufacturing process. Firstly, data exploration is carried out and it reveals several data challenges: (1) highly imbalanced data, (2) noisy data, (3) small sample size, and (4) component difference. Secondly, to address these challenges, statistical feature extraction is proposed to compress the pin-level dataset into component-level. Thirdly, a neural-net model with feeding imbalance control is established for Task 1. Fourthly, the random forests model is applied for both Task 2 and Task 3. Moreover, a threshold moving technique is proposed to optimize the threshold selection. Finally, the results show that our models achieved F1-scores of 0.44, 0.54 and 0.71 (average of 0.56) using the test dataset for the three tasks, respectively.

REFERENCES

Giordano, D., & Trevisan, M. (2022). "PHME Data Challenge". *European conference of the prognostics and health management society*.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5-32.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *the Journal of Machine Learning Research*, 12, 2825-2830.

Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing

machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1), 20-29.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321-357.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008, June). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)* (pp. 1322-1328). IEEE.

Provost, F. (2000, July). Machine learning from imbalanced data sets 101. In *Proceedings of the AAAI'2000 workshop on imbalanced data sets* (Vol. 68, No. 2000, pp. 1-3). AAAI Press.

Maloof, M. A. (2003, August). Learning when data sets are imbalanced and when costs are unequal and unknown. In *ICML-2003 workshop on learning from imbalanced data sets II* (Vol. 2, pp. 2-1).

BIOGRAPHIES

Haichuan Tang received a B.E. (in 2009), and Ph.D. (in 2015) both in Electrical Engineering from Southwest Jiaotong University, China. He also received Engineer Diploma in 2012 from Ecole Centrale de Nantes, France. He currently leads the AI lab of CRRC Academy. His main research interests are data-based fault diagnosis and prognosis.

Yin Tian received a B.E. degree in Electronic and Information Engineering, in 2009, and a Ph.D. degree in Traffic Safety Engineering, in 2015, both from Beijing Jiaotong University. He currently serves as deputy director of the Technology Research Department of CRRC Academy. He is conducting research regarding AI, big data, and their applications in manufacturing, logistics, and maintenance.

Junyan Dai is a Ph.D. student at Rutgers University. His research focuses on big data analysis for intelligent transportation problems. He received a B.S. degree in Computer Science from Rutgers University.

Yuan Wang obtained his Ph.D. (in 2019) and bachelor's degree (in 2014) at Southwest Jiaotong University, China. He is a cool guy who loves programming and algorithm design.

Jianli Cong is a Ph.D. student in Civil Engineering at Southwest Jiaotong University. His research focuses on railway inspection and sensing technologies.

Qi Liu received a B.E. degree in 2016, and M.Eng. Degree in 2018, both in Software Engineering, from Beijing Jiaotong University. She currently works at the AI Lab of CRRC Academy. Her research interests include Prognostic and Health Management in the field of rail transit and wind power.

Xuejun Zhao received his Ph.D. degree in Safety Science and Engineering from Beijing Jiaotong University. He has joined CRRC Academy as an algorithm engineer since 2020. His research interests mainly focus on signal processing algorithm development and algorithm acceleration based on multi-core computing platforms.

Yunxiao Fu received his Ph.D. in Transportation tool application engineering from Beijing Jiaotong University (Beijing, China), in 2017. His current research projects in AI Lab of CRRC Academy include developing intelligent control strategy of train in transit operation.

Deep learning representation pre-training for industry 4.0

Alaaeddine Chaoub¹, Christophe Cerisara², Alexandre Voisin³, and Benoît Iung⁴

^{1,3} *Université de Lorraine, CNRS, INRIA, LORIA, France*

alaaeddine.chaoub@loria.fr

christophe.cerisara@loria.fr

^{2,4} *Université de Lorraine, CNRS, CRAN, France*

alexandre.voisin@univ-lorraine.fr

benoit.iung@univ-lorraine.fr

ABSTRACT

Deep learning (DL) approaches have multiple potential advantages that have been explored in various fields, but for prognostic and health management (PHM) applications, this is not the case due to the lack of data in particular applications and also due of the absence of multiple DL-oriented benchmarks as in other fields, which limits the research in this area even though these types of applications will have a strong impact on the industrial world. To introduce the benefits of DL in this area, we should be able to develop models even when we have small data sets, to verify whether or not this is possible, in this thesis we explore the research direction of few shot learning in the context of equipment PHM.

Keywords— PHM, RUL prognostic, Deep learning, Few shot learning

1. CONTEXT

The context of this PhD is the industry of the future and more particularly the contribution of digitalization and Artificial Intelligence to predictive maintenance. Predictive maintenance is a strategy whose objective is to anticipate the failure (rather than to undergo it) with respect to the real state of a production system and forecasts on its operation. This anticipation thus makes it possible to minimize the drawbacks of traditional maintenance such as unexpected breakdowns interrupting production, lack of spare parts for repairs, to name but a few. This approach to maintenance, based on the digitalization of companies, uses the data collected to predict and forecast the evolution of degradation and propose the maintenance actions best suited to the current situation of the production system in order to anticipate failures by limiting unnecessary operations. Thus, the prognosis consists of evalu-

ating the Remaining Useful Lifetime (RUL) of a system, for example, a component, a machine, or even a production line. This is done by predicting the future state of health of the system up to its failure based on available past/present/future information, such as history, current operating data, but also future production planning and planned maintenance actions.

The objective of this PhD is therefore to propose deep learning models for the analysis and representation of available data to make a prognosis. Data analysis is essential because although the quantity of data available in this future industry context is often important, the data relevant for prognosis can be rare (infrequent event), of uncertain quality, unlabeled, partial, unbalanced... To address this issue, the originality of the approach to be followed in this PhD is based on the pre-training of deep learning networks. Indeed, in the field of deep learning, the unsupervised learning of representations allows to exploit all available data, annotated or not, and to extract generic information transferable to all target tasks of classification and prognosis. This makes it possible to considerably reduce the size of the learning dataset for the task at hand.

While such representations have already been successfully explored in the fields of image recognition in 2014, with convolutional architectures trained on ImageNet, and automatic natural language processing in 2018, with attention models trained in particular on word prediction tasks on the internet, they do not yet exist in the industrial field, which probably explains at least in part why 'the ImageNet moment' of the industry of the future has not yet taken place. The objective of this PhD is to contribute to the construction of such an industrial data representation model, by designing models and tasks that are not necessarily directly related to a target application, but that allows to efficiently encode the richest and most generic information possible on some underlying industrial processes, such as the degradation of mechanical parts over time.

AlaaeddineChaoub et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. RUL PROGNOSTIC

Data based approaches are more and more used for RUL prognostic due to their ability to model highly nonlinear, complex and multidimensional systems. A number of deep learning (DL) techniques have been deployed in order to learn the mapping from monitored system data to their associated RUL.

In this regard, we proposed a simple yet powerful model architecture for RUL prediction, the proposed model has an MLP-LSTM-MLP architecture trained in an end-to-end manner (Chaoub, Voisin, Cerisara, & Iung, 2021). Generally, Recurrent neural networks are often used for problems involving time series data, because of their ability to process information over time. However, LSTM cells are designed to capture time dependencies but they do not have the capacity to handle complex feature processing, which has led other works in the literature to perform this task manually before the learning phase. Conversely, MLP are well fitted to perform such a task. We thus propose to feed all of the raw inputs into an MLP before the LSTM layers. The MLP will be in charge of processing the raw inputs and learning a good representation of each time frame, while the LSTM shall capture the dependencies through time of frame sequences. Then, a final regression head, composed of another MLP, predicts the RUL from these temporally smoothed representations. The proposed method was tested on the public C-MAPSS dataset (Saxena, Goebel, Simon, & Eklund, 2008). Comparisons with several state-of-the-art approaches were performed, showing that our model outperforms the others for complex datasets with multiple OCs.

3. SMALL DATASET PROBLEM

In recent years, multiple deep learning approaches have been proposed for RUL prediction. However, these models are data demanding, which is a big drawback when it comes to real industrial PHM applications.

Relevant data for prognostics are often scarce, expensive to obtain, unbalanced. Indeed, several factors can lead to such a situation like:

- most of industrial system are quite reliable by design,
- preventive maintenance makes the occurrence of failure even more rare,
- despite the monitoring of the system and huge amount of available data most of them are in good operation state,
- labelling the data is not an easy process as it is usually a manual process and requires to explore the maintenance report,
- trials to obtain run-to failure process data cannot be implemented at the line level since failure usually takes long time.

When looking in the literature, The majority of the works do not face this problem because they work on benchmarks dedicated to the development of DL models like the C-MAPSS data set or other available data set in the NASA repository (Saxena et al., 2008) or dedicated laboratory tests that usually are not able to represent the complexity and variability of situation faced with real industrial application. Indeed, in a real industrial use case, we will most likely face the case where we have a very limited number of representative trajectories, which may lead to poor generalization and performance.

To overcome this problem of insufficient data sets, there is a sub-field of machine learning called "few-shot learning" which goal is to imitate the rapid learning ability of humans by being able to learn a new task with only a small number of labeled samples. This sub-area has received a lot of attention in recent years, multiple approaches and new benchmarks are proposed in this context.

3.1. Learning from few samples

Few-shot learning (FSL) is the problem of making predictions based on a limited number of samples (usually < 20). it can be used for regression and classification tasks. There are three main approaches in the litterature for FSL:

Data augmentation: These approaches aim to generate more samples from the few examples given, either by synthesizing new data using a generative model (Hariharan & Girshick, 2016; Iwana & Uchida, 2020), or using external knowledge or data (Jin & Rinard, 2020; Iwana & Uchida, 2020).

Metric learning: This family of approaches learns a nonlinear embedding in a metric space where a simple metric function is used to determine the output value of the new samples via proximity to the few labeled learning examples embedded in the same space. These approaches are widely used for few shot classification tasks (Vinyals, Blundell, Lillicrap, Kavukcuoglu, & Wierstra, 2016; Sung et al., 2017; Snell, Swersky, & Zemel, 2017).

Meta-learning: Also known as Learning-to-learn, These methods are trained on a set of episodes (few-shot tasks) instead of a set of object instances, with the motivation to learn a learning strategy that will allow effective adaptation to new such tasks using one or few examples (few-shot). Two big families of meta-learning methods exist in the literature, Gradient based meta learning, which goal is to find the optimal parameters of a model such that it can be easily fine-tuned on a new task (Finn, Abbeel, & Levine, 2017; Nichol, Achiam, & Schulman, 2018; Li, Zhou, Chen, & Li, 2017) and Metric meta learning approaches, which goal is combine the advantages of metric learning and meta learning (episodic learning) (Vinyals et al., 2016). These kind of approaches rely on having tasks that are close to the task at hand.

3.2. Fewshot learning possible directions for PHM

The approaches presented above are very promising. However, in the context of PHM applications, many circumstances restrain the methods we can try. Meta-learning and Metric learning approaches are difficult to apply when the data we have for episodic training or to train the embedding model, respectively, do not have identical number of input features, which is the case when we do not have the same sensors. Also, data augmentation can not work without having a strong prior on data distribution. Finally, whatever the approach chosen, having to merge data sets with large differences in the length of the sequences into the same approach is also a major problem to deal with.

An approach that can solve this problem in industry must be able to address all of the above challenges. During this period of the thesis, while studying the data sets available in the context of PHM, we are in the phase of adapting and assessing these three paradigms in order to propose an adapted approach that would make progress to solve these challenges and enable few-shot learning for PHM.

4. CONCLUSION

The lack of relevant industrial data for prognostic stands as barrier to achieving a more reliable and sustainable industry, while PHM of equipments has theoretically proven to be an approach to maximize profit and provide more safety for workers, its application to real-world data still remains a pressing question. During this thesis, we do research in the direction of Few shot learning, which could provide a practical solution that could be applied in real-world scenarios, with the goal of having a broad impact.

ACKNOWLEDGMENT

This work is part of the project AI-PROFICIENT which has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 957391. The Grid5000 computing resources have been used to partly train and evaluate the proposed models.

NOMENCLATURE

<i>PHM</i>	Prognostics and Health Management
<i>MLP</i>	Multi layer perceptron
<i>LSTM</i>	Long-short term memory
<i>OC</i>	Operating condition
<i>FSL</i>	Few shot learning

REFERENCES

- Chaoub, A., Voisin, A., Cerisara, C., & Iung, B. (2021, June). Learning representations with end-to-end models for improved remaining useful life prognostic. In *European Conference of the Prognostics and Health Management Society* (Vol. 6). Virtual event, Italy.
- Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR, abs/1703.03400*. Retrieved from <http://arxiv.org/abs/1703.03400>
- Hariharan, B., & Girshick, R. B. (2016). Low-shot visual object recognition. *CoRR, abs/1606.02819*. Retrieved from <http://arxiv.org/abs/1606.02819>
- Iwana, B. K., & Uchida, S. (2020). An empirical survey of data augmentation for time series classification with neural networks. *CoRR, abs/2007.15951*. Retrieved from <https://arxiv.org/abs/2007.15951>
- Jin, C., & Rinard, M. (2020). Learning from context-agnostic synthetic data. *CoRR, abs/2005.14707*. Retrieved from <https://arxiv.org/abs/2005.14707>
- Li, Z., Zhou, F., Chen, F., & Li, H. (2017). Meta-sgd: Learning to learn quickly for few shot learning. *CoRR, abs/1707.09835*. Retrieved from <http://arxiv.org/abs/1707.09835>
- Nichol, A., Achiam, J., & Schulman, J. (2018). On first-order meta-learning algorithms. *CoRR, abs/1803.02999*. Retrieved from <http://arxiv.org/abs/1803.02999>
- Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *2008 international conference on prognostics and health management* (pp. 1–9).
- Snell, J., Swersky, K., & Zemel, R. S. (2017). Prototypical networks for few-shot learning. *CoRR, abs/1703.05175*. Retrieved from <http://arxiv.org/abs/1703.05175>
- Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. S., & Hospedales, T. M. (2017). Learning to compare: Relation network for few-shot learning. *CoRR, abs/1711.06025*. Retrieved from <http://arxiv.org/abs/1711.06025>
- Vinyals, O., Blundell, C., Lillicrap, T. P., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. *CoRR, abs/1606.04080*. Retrieved from <http://arxiv.org/abs/1606.04080>

Physics Informed Self Supervised Learning For Fault Diagnostics and Prognostics in the Context of Sparse and Noisy Data

Weikun Deng, Khanh T. P. Nguyen, and Kamal Medjaher

*Laboratoire Génie de Production, LGP, Université de Toulouse, INP-ENIT, 47 Av. d'Azereix, 65016, Tarbes, France.
weikun.deng@enit.fr, nguyenv@enit.fr, kamal.medjaher@enit.fr*

ABSTRACT

Sparse & noisy monitoring data leads to numerous challenges in prognostic and health management (PHM). Big data volume but poor quality with scarce healthy states information limits the performance of training machine learning (ML) and physics based failure modeling. To address these challenges, this thesis aims to develop a new hybrid PHM framework with the ability to autonomously discover and exploit incomplete implicit physics knowledge in sparse & noisy monitoring data, providing a solution for deep physics knowledge-ML fusion by physics-informed machine learning algorithms. In addition, the developed hybrid framework also apply the self-supervised learning paradigm to significantly improve the learning performance under uncertain, sparse, and noisy data with lower requirements for specialist area knowledge. The performance of the developed algorithms will be investigated on the sparse and noise data generated by simulation data sets, public benchmark data sets, and the PHM platform to demonstrate its applicability.

Keywords—Prognostic and health management; Sparse & noisy data; Hybrid framework; Physics informed machine learning; Self-Supervised Learning.

1. MOTIVATION AND RESEARCH PROBLEM STATEMENT

Prognostics and health management (PHM) plays a constructive role in ensuring the real-time health assessment of a system under its actual working conditions as well as the prediction of its future state based on up-to-date information (N. Kim, An, & Choi, 2017). Two mainstream methods, which are mainly used are Machine Learning (ML) and Physics-based methods (PBM). ML is proficient at automatically extracting features from data and building relationships between features based health indicators and system states. However, as a data-hungry and black-box method. ML meets dilemmas in processing sparse & noisy data. The pervasive monitor-

ing instrument costs, the high run-to-failure operation costs, and the lack of data label are objective conditions that create sparse/noisy data that is insufficient for ML to learn a meaningful knowledge representation. Besides, PBM represent the degradation mechanisms by observing failure phenomena and then establishing mathematical equations or numerical laws, with the ability to infer hidden states from a limited sample (Chao, Kulkarni, Goebel, & Fink, 2019). However, modern engineering systems have simultaneous non-linear interactions between their subsystems and their environment. Failure mechanisms and degradation processes are difficult to identify. With incomplete failure cognition, implementing detailed parametric or numerical degradation models for these systems in sparse & noisy data is challenging.

These challenges prompt PHM techniques into a hybrid framework. Hence, this thesis aims to explore the combination of PBMs and ML by physics informed machine learning (PIML). Providing a deep model & data-driven embedding fusion solution to assist trustworthy PHM deployment in “small data, small laws” contexts. The developed framework is hoped to be trained in self supervised learning training (SSL) paradigm to build the ability to autonomously discover and exploit implicitly incomplete physics knowledge in sparse noisy monitoring data.

2. NOVELTY AND SIGNIFICANCE RELATIVE TO THE STATE OF THE ART

To the best of our knowledge, the research about SSL-PIML hybrid framework is scarce, most of them are derived from reconstructive recognition of image data in the medical field, and physics-based loss functions are designed to test the effectiveness of the feature extractors in pretext (Yaman et al., 2020; Martín-González et al., 2021). A brief review of advanced research on PIML and SSL in PHM is performed. The bibliometrics results from Citespace analysis for 185 PIML hybrid methods -related and 35 SSL hybrid methods related papers are presented in Fig.1. In PHM field, the rotating machinery, grid, production lines, batteries, and materials are the main application scenarios of SSL and PIML techniques while the anomaly detection, fault diagnosis, and RUL pre-

WeiKun Deng et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

diction are their core objectives.

On one hand, the PIML methods have risen and are attracting more attention since 2017. According to the way of integrating physics knowledge in the ML pipeline, the PIML methods can be categorized into three groups: 1) Physics informed input space, 2) Physics informed structure, and 3) Physics informed loss function (Karniadakis et al., 2021). According to the Fig.1, the research has shifted from the integration of features and rules (expert systems) to the integration of algorithms structure and parametric models (physics informed neural network (Karniadakis et al., 2021)). It suggests that researchers seek to create an augmented input feature and a physics informed (PI) derivation process that can be interpretable (S. W. Kim, Kim, Lee, & Lee, 2021). In this process, a variety of refined laws and analytic relations such as linear damage accumulation laws, crack extension formula are incorporated by the different methods such as neural networks (Viana & Subramanian, 2021), Gaussian processes (Cury, Ribeiro, Ubertaini, & Todd, n.d.). Besides, the research related to embedding the partial differential equations representing system behaviors into ML models is gradually becoming a popular method. The essence of PIML is to introduce physics constraints to ML data processing process. Its drawback is the high requirement of physics domain knowledge because the incorporation methods still relies heavily on manual designed explicit knowledge with parsed form.

On the other hand, SSL methods mainly focuses on mining its own supervised information from large-scale unlabelled monitoring data using an auxiliary task (pretext), and training ML with this constructed supervised information to build valuable representations for downstream detection, diagnostic, and prediction tasks. It is clear in Fig.1 that SSL methods in PHM are in the stage of self-supervised feature engineering. They focus on signal reconstruction and feature extraction through principal component analysis (PCA) (Wang, Qiao, Zhang, Yang, & Snoussi, 2020), Deep Clustering and Auto-encoder (Zhang, Chen, He, & Zhou, 2022), Generative Adversarial Network (Ding, Zhuang, Ding, & Jia, 2022). In these studies, self-supervised (SS) features construct bounds for different health states by fine-tuning valuable representations for downstream tasks, such as bounds for reconstruction error as a normal-abnormal watershed and bounds for similarity as a distinguishing representation for different fault states. Particularly, SSL based RUL predictions are rarely studied. Moreover, only generative schemes are widely used compared to the other SSL architectures, e.g., contrastive or generative-contrastive strategies.

In summary, the focus of the hybrid framework proposed in this study is autonomously incorporating the implicit incomplete physical knowledge into ML, under sparse/noisy monitoring data. It is an issue that is hardly mentioned in existing studies but indeed needs to be addressed by original and innovative research in the development of PHM without delay.

3. WORK PROGRESS AND FUTURE DIRECTION

Motivated by the philosophical concept of “constructivism learning”, it is hoped to build PIML-SSL hybrid framework based on conformity and assimilation. In conformity, ML transforms the original data-driven reasoning process by incorporating physics constraints. In assimilation, ML trains feature extractors in self-supervised way for downstream PHM tasks without changing the PIML framework. Currently, the literature review has completed and based on it, this thesis is at the beginning of the methodological development. Particularly, the developed hybrid framework using PI-SSL paradigm consists of the following techniques in Fig.2:

a) Knowledge - ML module inter-conversion mechanisms

The inter-conversion mechanisms are dedicated to embedding the mathematical relations, i.e., Input-output (IO) model (analytical function) or physics operator (differential relationships) of the failure to a part of the ML calculation diagram in layer functions, regression formulas, coefficient distribution, etc. Based on generic mathematical relations, ML will infer uncertain parameters and automatic search for hidden representations of the possible formation of degradation relations for these units of embedded physical knowledge.

b) Physics informed metric learning

It aims to establish boundaries metric distances for failure states and the corresponding HI. This enables the ML's results to respect the basic physics consistency such as physics-informed similarity, principle of cumulative energy dissipation for wear behavior, etc. In particular, distance measures between different health states based on comparative learning will be investigated in depth.

c) Boundary condition exploration pretext task design

In SSL, the inter-conversion mechanisms helps to establish a downstream data-driven health indicator (HI) according to PHM tasks. In detail, an appropriate PI computational structures will be constructed to complete the assimilation process, e.g., Siamese, Codec, Graph, etc. These structures seek to maximize the difference between the boundaries of different health states while satisfying physics consistency. Through training on a “pretext task”, these structures automatically generates pseudo labels. Their parameters are frozen as a supervised feature extractor, connecting with the different functional ML module in the fine tuning process when it is transferred into the downstream PHM tasks.

d) Hybrid framework design and validation

Based on the previous research, we will construct the PI-SSL hybrid framework. The quantitative and sensitivity analyses of data quality on the its performance will be performed to probe the lower limit of tolerance and upper limit of its applicability in sparse & noise and incomplete physics knowledge. Relevant metrics on sparsity, noise, and knowledge completeness will be defined and quantified through masking and selective cropping of public data sets, mechanistic models, and experimental data. The influence of the above indicators to

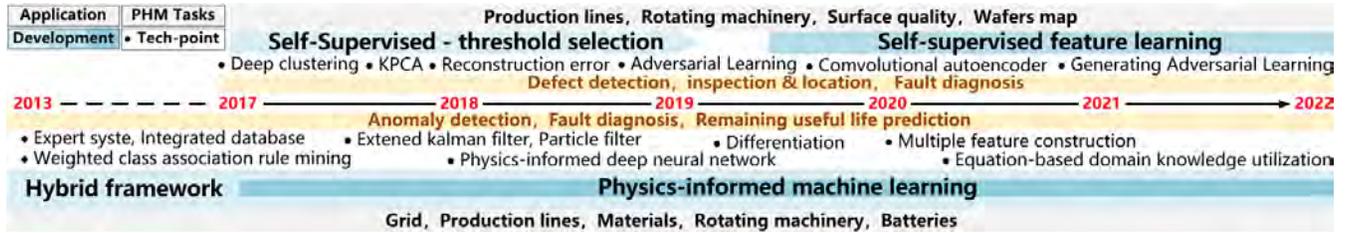


Figure 1. SSI and PIML tech-development analysis

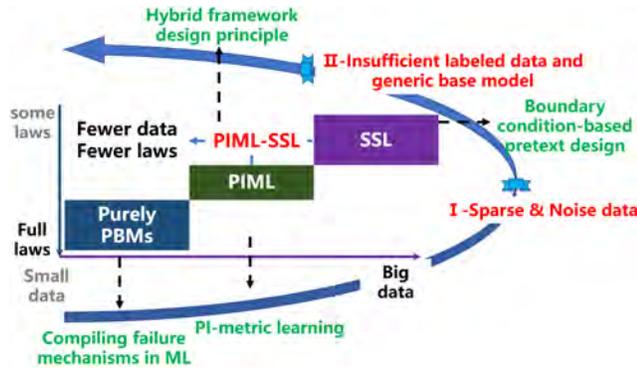


Figure 2. Technology road-map for the thesis.

varying degrees under different working conditions will be studied deeply in the rotating machine PHM test platform. The results from different data sources will be compared to find the difference in their feature representation.

4. DISCUSSION ON THE APPLICATIONS

This thesis develop a PIML hybrid framework equipped with SSL training paradigm for fault diagnostics and prognostics purposes in the context of sparse & noisy data with incomplete and implicit failure knowledge. It design the physics informed operator or ML module to completes the seamless methods integration, establishing fault boundary metric distance in the objective function to improve the physics consistency as well as to reduce the data dependency of ML. In fact, the need of large labeled and high-quality data is too difficult or costly to satisfy. Meanwhile, the ability to correctly interpret the output of a PHM model is essential in high-value devices. The developed hybrid PHM framework with physics consistency and excellent exploitation of sparse & noise data allows better understanding of the system state and maintenance supports. Its algorithms potentially extended to the large-scale and low-cost deployment.

REFERENCES

Chao, M. A., Kulkarni, C., Goebel, K., & Fink, O. (2019). Hybrid deep fault detection and isolation: Combining deep neural networks and system performance models.

arXiv preprint arXiv:1908.01529.

Cury, A., Ribeiro, D., Ubertini, F., & Todd, M. D. (n.d.). *Structural health monitoring based on data science techniques*. Springer.

Ding, Y., Zhuang, J., Ding, P., & Jia, M. (2022). Self-supervised pretraining via contrast learning for intelligent incipient fault detection of bearings. *Reliability Engineering & System Safety*, 218, 108126.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6), 422–440.

Kim, N., An, D., & Choi, J.-H. (2017, 01). Prognostics and health management of engineering systems..

Kim, S. W., Kim, I., Lee, J., & Lee, S. (2021). Knowledge integration into deep learning in dynamical systems: an overview and taxonomy. *Journal of Mechanical Science and Technology*, 1–12.

Martín-González, E., Alskaf, E., Chiribiri, A., Casaseca-de-la Higuera, P., Alberola-López, C., Nunes, R. G., & Correia, T. (2021). Physics-informed self-supervised deep learning reconstruction for accelerated first-pass perfusion cardiac mri. In *International workshop on machine learning for medical image reconstruction* (pp. 86–95).

Viana, F. A., & Subramaniyan, A. K. (2021). A survey of bayesian calibration and physics-informed neural networks in scientific modeling. *Archives of Computational Methods in Engineering*, 28(5), 3801–3830.

Wang, T., Qiao, M., Zhang, M., Yang, Y., & Snoussi, H. (2020). Data-driven prognostic method based on self-supervised learning approaches for fault detection. *Journal of Intelligent Manufacturing*, 31(7), 1611–1619.

Yaman, B., Hosseini, S. A. H., Moeller, S., Ellermann, J., Uğurbil, K., & Akçakaya, M. (2020). Self-supervised learning of physics-guided reconstruction neural networks without fully sampled reference data. *Magnetic resonance in medicine*, 84(6), 3172–3191.

Zhang, T., Chen, J., He, S., & Zhou, Z. (2022). Prior knowledge-augmented self-supervised feature learning for few-shot intelligent fault diagnosis of machines. *IEEE Transactions on Industrial Electronics*.

A Novel Way to Apply Transfer Learning to Aircraft System Fault Diagnosis

Lilin Jia¹, Cordelia Mattuvarkuzhali Ezhilarasu¹, and Ian Jennions¹

¹*IVHM Centre, Cranfield University, Bedfordshire, MK43 0AL, United Kingdom*

lilin.jia@cranfield.ac.uk

c.m.Ezhilarasu@cranfield.ac.uk

i.jennions@cranfield.ac.uk

Keywords: fault diagnosis, transfer learning, aircraft systems, IVHM, condition-based maintenance

ABSTRACT

In recent years, transfer learning as a method that solves many issues limiting the real-world application of conventional machine learning methods has received dramatically increasing attention in the field of machine fault diagnosis. One major finding from an initial literature review shows that the majority of the existing research only focus on the transfer of diagnostic knowledge between various conditions of the same machine or different representation of similar machines. The primary goal of the current work is to seek a way to apply transfer learning to distinct domains, thereby expanding the boundary of transfer learning in the fault diagnosis field. In particular, attempts will be made to explore ways of transferring knowledge between diagnostic tasks of different aircraft systems. One promising method to help achieving this goal is transfer learning by structural analogy, since this method is capable of extracting high-level structural knowledge to apply transfer learning between seemingly unrelated domains, similar to the scenarios of transfer between different aircraft systems.

1. MOTIVATION AND RESEARCH PROBLEM

Transfer learning has recently been a popular research topic in the field of Intelligent Fault Diagnosis (IFD). As Lei, Yang, Jiang, Jia, Li, and Nandi (2020) pointed out, transfer learning is the promising method to “expand IFD from academic research to engineering scenarios”. Its capacity to overcome the various factors that render conventional machine learning algorithm inaccurate or inapplicable in the diagnosis of real-case machines had recently attracted a great deal of attention in the academic world, which is demonstrated by the dramatic increase in the number of publications on applying transfer learning to IFD in the recent three years. Therefore, this work aims to investigate this exciting new research trend and to develop a novel transfer learning method to apply to aerospace systems.

The initial literature review had discovered that, in the field of IFD, most existing research focus on applying transfer learning only between similar tasks, in which low-level similarities such as the data structure and the physical parameters are required to be the same. This work will attempt to expand the boundary of transfer learning method by addressing more distinct domains of transfer, where low-level similarities can no longer be depended on. Specifically, the research problem was determined to be designing a novel method that achieves knowledge transfer between the diagnosis of different aircraft systems.

2. LITERATURE REVIEW

To collect relevant literature, a keyword search of “transfer learning” with “machine fault diagnosis” was conducted on Scopus in November 2021, which returned 412 publications. A similar pattern was discovered when analysing the research trend of the first 70 and second 70 entries, hence these 140 entries were decided to be adequate to represent an overview of the field of applying transfer learning to IFD.

2.1. Research motives

The top four motives for conducting the research as stated in the publications are: 1) to solve small sample problems; 2) to solve the lack of labelled samples; 3) to overcome the shortage of faulty data; 4) to adapt to the data distribution discrepancy over the domains of concern. This is consistent with the theory that transfer learning is fundamentally introduced to overcome the limitations of conventional machine learning methods (Yang, Zhang, Dai, and Pan, 2020).

In the field of IFD, conventional machine learning methods would only perform accurately if the training data met all four requirements simultaneously: 1) are in sufficient quantity; 2) have sufficient labelling; 3) are balanced; 4) display the same distribution pattern as the testing samples.

However, real-world machines rarely produce such ideal training samples, hence it has motivated researchers to apply transfer learning to IFD.

2.2. Application fields and validating examples

The fields of application showed a high concentration on bearings and gearboxes, which respectively took up 57% and 15% of the research studied. A further inquiry into the validating examples was conducted in this work and discovered that 85% of the bearing research referred to, in their method validating process, the same bearing dataset published by Case Western Reserve University (CWRU). The lack of diversity in the fields of application and the high dependency on the same validating dataset raised the concern on whether many existing transfer learning-based IFD algorithms could have good generalization capacity over alternative applications.

The other 28% of research focused on various applications such as transformers, wind turbines, induction motors and so on. Since this work aims to apply transfer learning to aerospace, a search for all aerospace-related research within the 140 publications, and all other entries outside the 140 results on Scopus, was performed. This found examples only existed in the fault diagnosis of spacecraft attitude control system, aero-engine gas path, electromagnetic actuators, aircraft fuel pump, UAV inertial sensors, quadrotor, and commercial flight data. Therefore, there are numerous opportunities to apply transfer learning in aerospace IFD, and to do it across the diagnosis of different aircraft systems is among such research gaps.

2.3. Domains of transfer

Another important aspect of research gathered for the 140 papers is the selection of the source and target domains between which the transfer happens. Overall, most research only discussed transfer between similar domains. As statistics showed, 55% of the research discussed transfer between the diagnosis of the same machine under various working conditions that either involve a selection or a combination of varied load, rotational speed or degradation level. Another 27% of the research was on the transfer between different representation of the same machine - either between a lab-scale test rig and the real-size machine, between simulation and real data or between slightly different sub-type of the same machine.

Although the majority of transfer learning work in IFD only considered transfer between similar domains, some attempts were made to push the boundary of transfer. For instance, one common assumption when applying transfer learning to IFD is that the target domain label space must be either identical to, or a subset of, the source domain label space (Lei et al., 2020). Li, Huang, He, Wang, Li, and Li (2020) attempted to break this assumption by introducing a new fault type in the target task that is not present in the source

domain samples. Li et al. (2020) chose both the source and target domain samples from the CWRU dataset, but the target domain samples were taken at different load and speed with outer race fault chosen as the additional fault type than the source domain. Li et al. (2020) managed to handle the difference in the label spaces by introducing a pseudo decision boundary after the feature extraction stage that distinguishes the new emerging fault from known fault classes. Pushing the transfer boundary ever further, Deng, Huang, Du, Li, Zhao, and Lv (2021) considered the transfer on different machines (TDM) by partial transfer with different faults. Deng et al. (2021) designed a double-layer attention based adversarial network which essentially conduct domain adaptation in a discriminative way to minimize the negative effect from irrelevant source data. This method by Deng et al. (2021) demonstrated the ability to transfer diagnostic knowledge between different types of bearings under different working condition with different damage modes and display different damage characteristics, all at the same time.

Inspired by these attempts, this work has proposed the idea to push the boundary of transfer even further by seeking a way to transfer diagnostic knowledge not just between different machines, but between distinct machines, such as different aircraft systems.

3. PROPOSED APPROACH

3.1. The inspiration from the history of transfer learning

Transfer learning, being a machine learning paradigm as it is seen today, has a historical root in transfer of learning in the field of cognitive science (Yang et al., 2020). Transfer of learning studies the phenomenon that “humans can draw on the past experience to solve current problems very well (Yang et al., 2020)”.

One way how transfer of learning evolved to transfer learning originated from the biological aspect of human learning. As Bozinovski (2019) pointed out, the earliest mathematical model on transfer learning was based on neural networks. This finding explains why most transfer learning method commonly seen are heavily network-based.

However, current assumptions of the common transfer learning methods restrict applications to different but similar tasks, which cannot accommodate the goal of this work. As a result, this work had focused on how the other aspect of transfer of learning, the conceptual aspect of human learning, might have contributed to transfer learning. The reason for this shift in focus is that humans are capable of resolving high level similarities to learn seemingly unrelated tasks (Yang et al. 2020). Transfer learning that captures this high-level transfer is in line with the goal to achieve transfer between more distinct domains. The next sub-section gives one method that bears some promising features to this work’s research problem.

3.2. Introduction to transfer learning by structural analogy

Transfer learning by structural analogy is a type of relation-based transfer learning method. Unlike other relation-based transfer learning method, it is unique in the way that analogues are found in the domains of interest as the output of the algorithm. Consequently, analogues found in the seemingly unrelated domains work as the bridge that achieves the knowledge transfer which is otherwise considered impossible.

Wang and Yang (2011) designed such algorithm that achieves transfer where the source and target domains have completely different representation spaces. Since there is the absence of low-level similarities for such transfer scenario, Wang and Yang (2011) first introduced a mapping of the features from both domains into the Reproducing Kernel Hilbert Space (RKHS), where the structural dependencies of the features can be estimated. By maximizing the dependencies between the features and the labels in both domain as well as between the features across the two domains, features that bear analogical value, and also help to establish the correct labels, are identified (Wang & Yang, 2011). Wang and Yang (2011) used a medical diagnosis dataset to validate the analogy found by the algorithm, where after the algorithm identified ten pairs of analogical terms in the diagnosis of cardiovascular diseases and respiratory tract diseases, by treating the analogues in both domains equivalent, the classifier trained in the source task yielded 80.5% accuracy in the target task.

This work considers this algorithm as a promising approach for the problem of transferring diagnostic knowledge between different aircraft systems, since abundant structural relations exist in aircraft system data. Adaptation of the method will be necessary for this application. For example, if analogues of features in the symptom vectors of two aircraft systems can be found, by treating them as equivalents, transfer of a pre-trained classifier could be effectively implemented.

4. APPLICATIONS AND CONTRIBUTIONS OF THE WORK

Applying transfer learning to aircraft system fault diagnosis is a promising way to enhance the smart diagnosis process. Upon successful transfer of diagnostic knowledge across different aircraft systems, the robustness and accuracy of the diagnostic model are expected to be benefited.

This work will contribute to the progress in the field of condition-based maintenance (CBM) and integrated vehicle health management (IVHM). If the ambition of transfer learning between distinct machines is achieved, the common belief in the boundary of transfer learning would be largely expanded.

5. WORK IN PROGRESS

Surrounding the thesis on applying transfer learning in the field of aircraft fault diagnosis, the author has only reached the last stage of his initial literature review and is currently drafting a literature review paper. The paper will discuss the importance of adopting machine learning and transfer learning method in aircraft fault diagnosis, an investigation into the transfer learning method, an overview of the existing research of applying transfer learning to IFD, and the proposed method to resolve the research problem identified in the research gap. However, there has not been any experimental work to execute the method proposed at the current stage.

6. CONCLUSION

Focusing on the topic of applying transfer learning to the diagnosis of aircraft, this work has discovered research gaps in applying the transfer learning method to aircraft systems and between transfer domains that are more distinct. Inspired by the investigation into the history and the whole spectrum of transfer learning methods, a possible route to address the main research problem of transferring diagnostic knowledge between different aircraft systems is identified to be applying transfer learning by analogy, which could potentially push the boundary of transfer learning in IFD.

REFERENCES

- Bozinovski, S., (2019). Reminder of the First Paper on Transfer Learning in Neural Networks, 1976. *Informatica*, vol. 44 (3), pp. 291-302. doi:10.31449/inf.v44i3.2828
- Deng, Y., Huang, D., Du, S., Li, G., Zhao, C., & Lv, J. (2021). A Double-Layer Attention Based Adversarial Network for Partial Transfer Learning in Machinery Fault Diagnosis. *Computers in Industry*, vol. 127. doi:10.1016/J.COMPIND.2021.103399
- Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., & Nandi, A. K. (2020). Applications of Machine Learning to Machine Fault Diagnosis: A Review and Roadmap. *Mechanical Systems and Signal Processing*, vol. 138. doi:10.1016/J.YMSSP.2019.106587
- Li, J., Huang, R., He, G., Wang, S., Li, G., & Li, W. (2020). A Deep Adversarial Transfer Learning Network for Machinery Emerging Fault Detection. *IEEE Sensors Journal*, vol. 20(15), pp. 8413–8422. doi:10.1109/JSEN.2020.2975286
- Wang, H., & Yang, Q. (2011). Transfer Learning by Structural Analogy. *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence* (513-518), August 7-11, San Francisco.
- Yang, Q., Zhang, Y., Dai, W., & Pan, S. J. (2020). *Transfer Learning*. Cambridge: Cambridge University Press.

The Application, Utility and Acceptability of Data Analytics in Safety Risk Management of Airline Operations

Washington Mhangami, Stephen King, David Barry

Cranfield University, Bedfordshire, MK43 OAL, United Kingdom

washington.mhangami@cranfield.ac.uk s.p.king@cranfield.ac.uk d.jbarry@cranfield.ac.uk

Abstract One area the aviation industry is grappling with is the quantification of the probability of occurrence of safety incidents. Currently, aviation professionals involved in safety risk management mostly rely on collective experience to determine probability of incident occurrences and apply it to the International Civil Aviation Organisation (ICAO) matrix or equivalent to evaluate the risk. A number of limitations linked to the use of risk matrices will be explored in this paper. It is the aim of this paper to explore statistical methods that can be used to determine the probability of safety occurrences and come up with an algorithm that can be used by airlines using available safety data. The novelty of this research is that it combines the exploration of use of statistical techniques to quantitatively assess risk using Flight Data Monitoring (FDM) and other data, with acceptability of Safety Risk Management (SRM) data analytics by operational personnel. The paper also explores the contributory factors leading to the reluctance of operational personnel to use data analytics to inform their risk assessments despite the increasing availability of operational data and advancement in technology.

1. Motivation and research problem statement

The research idea stems from the perceived reluctance of the aviation industry to apply data analytical tools to improve safety risk assessment in Safety Management Systems (SMS) despite the availability of data from flight data recorders and safety reports. Safety risk is defined by ICAO (2018) as “The predicted probability and severity of the consequences or outcomes of a hazard”. In order to determine the safety risk index, probability and severity scores are combined in an alphanumeric format. Many airlines use the ICAO Safety Management Manual (2018) recommended matrix shown in figure 1, in their risk assessments.

Washington Mhangami et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Safety Risk		Severity				
		Catastrophic A	Hazardous B	Major C	Minor D	Negligible E
Frequent	5	5A	5B	5C	5D	5E
Occasional	4	4A	4B	4C	4D	4E
Remote	3	3A	3B	3C	3D	3E
Improbable	2	2A	2B	2C	2D	2E
Extremely improbable	1	1A	1B	1C	1D	1E

Figure 1. Safety Risk Matrix. Source: ICAO Safety Management Manual, 4th Edition

ICAO (2018), defines probability as “the likelihood that a safety consequence or outcome will occur”. The distinction between ‘improbable’ and ‘extremely improbable’ as an example, in practical terms, is vague and unhelpful, thus exposing some weaknesses in this matrix. While enhancements have been made in some organisations to adapt this matrix to suit their operations, the current matrix format (Figure 1) is still in use by many airlines and endorsed by numerous competent authorities.

There have been many criticisms to the use of safety risk matrices. Cox (2008) asserts that risk matrices have poor resolution and can incorrectly assign higher qualitative ratings to a risk of relatively smaller value leading to suboptimal resource allocation. Cox further argues that matrices consist of ambiguous inputs and outputs. Barry (2021) also argues that there has not been a lot of research looking into their validity, effectiveness and also evaluation of their performance in improving risk management decisions. Probabilities of occurrence require subjective interpretation and different users may obtain differing ratings of the same quantitative risk depending on their operational experience and national culture. Hubbard (2009) points out that the approaches that aviation organisations are using to manage risk lack quantitative analysis. Consequently,

organisations are most likely to come up with ineffective strategies which might worsen the risk situation.

Lishuai, Harisman, Palacios and Welsch (2016: p.1) affirm that “modern aircraft systems have become increasingly complex to a degree that traditional analytical systems have reached their limits”. Current methods are tailored to detect hazardous behaviours on parameters that have been pre-defined and they miss vital operational risks that are unlisted or unknown.

Risk assessment in big organizations with multi-operational domains is becoming increasingly challenging. Employing an effective method along with realistic pair comparisons taking opinions of organisational experts and removing the inherent bias in their inferences is problematic. It is becoming clear that traditional two-dimensional risk assessments to identify hazards and safety deficiencies lack the required sophistication to deal with increasingly complex airline operations (Rezai & Borjalilu, 2018). This is further echoed by Mauro and Bashi (2009) who highlight that “many risk assessment heuristics and displays can yield misleading and sometimes mathematically incongruous assessments”.

Airlines receive a lot of data from Flight Data Recorders (FDR) and Quick Access Recorders (QAR). In his research paper, “Estimating runway veer-off risk using a Bayesian network with flight data”, Barry (2021) argues that risk assessments in airline operations are mostly qualitative in nature and this is despite the availability of large amounts of data from programmes such as Flight Data Monitoring (FDM), employee safety reporting systems and Flight Operations Quality Assurance (FOQA).

A number of risk assessment methods such as Bow tie diagrams, The Airline Risk Management Solutions (ARMS), Safety Issue Risk Assessment are being used by some airlines to improve risk assessment in Safety Management Systems. While they are an improvement to simple matrices, they unfortunately still rely on some subjective assessment.

The safety record of civil aviation is unrivalled but if it is to be improved, the airline industry should transition towards a more proactive and potentially predictive approach which anticipates and mitigates operational risks before unwanted events occur. It is the aim of this paper to explore statistical methods that can be used to determine the probability of safety occurrences and develop an algorithm that can be used by airlines using available safety data.

2. Novelty and significance relative to the state of the art

ICAO Safety Management Manual (2018) mentions that “the level of detail and complexity of tables and matrices should

be adapted to the particular needs and complexities of each organisation”. This guidance is vague and can be interpreted in various ways by airlines. The manual recommends organisations to include both quantitative and qualitative criteria but it only gives examples of the latter. Arguably, this is the reason why most organisations are using the qualitative criteria.

From the 1st of January 2005, airlines that operate aircraft with a maximum take-off mass in excess of 27 tonnes are required by ICAO to have a FDM program. This program is good in that it highlights occurrences of a non-standard, abnormal or unsafe nature. The biggest challenge that airlines are facing is the translation of information of these unsafe occurrences into a useful measure of risk. It is vital for researchers to come up with novel ways of detecting anomalies in data automatically without the need for predefinition.

Statistical techniques have a role to play here as they have a range of practical applications to detect unusual events and abnormalities in terms of pre-defined limits and randomness of occurrence data. A few tentative statistical suggestions are given in UK Civil Aviation Publication (CAP) 739 but no further explanation is given on how the statistical methodologies can be incorporated into risk management.

This research explores the potential of using statistical techniques to come up with an algorithm that can be used on FDM occurrence data to inform a quantitative approach of safety risk probability of operations. The algorithm will have the ability to accommodate other qualitative data available from airline data sources to complement quantitative FDM data. This should significantly increase the reliability of determining probabilities of occurrence in airline operations.

Existing risk assessment methodologies evidently show that there is a reluctance to use technology to enhance quantitative risk assessment. This research is going to be underpinned by the Technology Acceptance Model (TAM) shown in figure 2 to try and understand this reluctance. This theory was specifically designed to assess an individual’s likelihood of accepting technology. This research will test whether the theory holds true in the use of data analytics in airline safety risk management processes. The fast speed of technology advancement in data analytics is challenging and its effects need to be explored. The research will also look into individual versus organisational (firm level) acceptance as well as velocity of the environment. The TAM model is going to be used as a general framework to investigate the factors that influence airlines to adopt data analytics in safety risk management of their SMS.

In their paper, “Understanding the usage, modifications, limitations and criticisms of Technology Acceptance

Model”, Malatji, Van Eck and Zuva (2020) discuss a number of limitations of the model. This research will also endeavour to improve the model in light of the cited shortcomings and more.

The model is shown in figure 2 below:

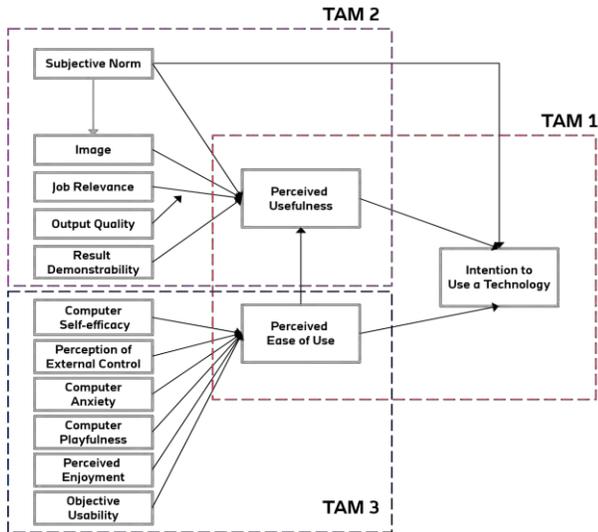


Figure 2. TAM 1, 2 & 3 – Simplified omitting moderators, Davis (1989) Venkatesh and Davis (2000) Venkatesh & Bala (2008). Source: Innovation Acceptance Lab

The novelty of this research is that it combines the exploration of use of statistical techniques to quantitatively assess risk using FDM and other data, with acceptability of SRM data analytics by operational personnel.

3. Discussion of the applications and the contribution of the work

The research will develop an algorithm that can be used by operational personnel to determine the probability of safety occurrence in airline operations. The study will also determine the factors which affect technology acceptance in an aviation industry which is highly dynamic and innovative.

4. Approach and proposed experiments (where appropriate)

A mixed methodology is proposed for this research. A quantitative approach is going to look into how occurrence data from a series of Airbus mid-range aircraft (A319, A320 and A321) flight data recorders and quick access recorders from the same operator. The data can be analysed and probabilities of occurrence determined using quantitative modelling. The resulting probabilities for significant events will be compared with those determined by the airline’s safety management team in their safety reports. The above-mentioned aircraft were flying European routes and standard

operating procedures were common across the different types. The flight recorder data and safety reports cover a 10-year period. The data analysis process will start in five months’ time.

A qualitative approach will be primarily carried out to get a better understanding of the factors which contribute to technology acceptance and also the evidently reluctance of aviation safety specialists to use quantitative methods of risk assessment. Interviews and questionnaires will be used to gather information from safety personnel involved in safety risk analysis and mitigation.

REFERENCES

Barry, D. J. (2021). Estimating Runway Veer-off Risk Using a Bayesian Network with Flight Data. *Transportation Research Part C: Emerging Technologies* 128:103180. doi: 10.1016/j.trc.2021.103180

Civil Aviation Authority (CAA) (2013). CAP 739 - Flight Data Monitoring. Available at: <http://publicapps.caa.co.uk/docs/33/CAP739.pdf> (Accessed: 8 June 2022).

Cox, L.A. (2008). What’s wrong with risk matrices? *Risk Analysis*. 28 (2), 497–512 doi: 10.1111/j.1539-6924.2008.01030.x.

Hubbard, D.W. (2009). *The failure of risk management*. Hoboken, NJ: John Wiley & Sons, Inc

International Civil Aviation Organisation (ICAO) (2018). Safety Management Manual (SMM). 4th ed. International Civil Aviation Organisation.

Li, L., Hansman, R.J., Palacios, R., & Welsch, R. (2016). Anomaly detection via a Gaussian Mixture Model for flight operation and safety monitoring. *Transportation Research Part C: Emerging. Technologies*. 64 (Supplement C), 45–57.

Malatji, W.R., Van Eck, R., & Zuva, T. (2020) Understanding the Usage, Modifications, Limitations and Criticisms of Technology Acceptance Model (TAM). *Advances in Science, Technology and Engineering Systems Journal*, vol.5, no. 6, 2020, pp.113-17. doi:10.25046/aj050612

Rezaei, M., & Borjalilu, N. (2018). A dynamic risk assessment modeling based on fuzzy ANP for safety management systems. *Aviation*, 22(4), 143-155. doi:10.3846/aviation.2018.6983

Venkatesh, V., & Hillol, B. (2008). ‘Technology Acceptance Model 3 and a Research Agenda on Interventions’. *Decision Sciences* 39(2):273–315. doi: 10.1111/j.1540-5915.2008

Diagnosis and fault-tolerant control for a multi-engine cluster of a reusable launcher with sensor and actuator faults

Renato MURATA^{1,2}, Louis THIOULOUSE¹, Julien MARZAT¹, H el ene PIET-LAHANIER¹, Marco GALEOTTA², Fran ois FARAGO²

¹ DTIS, ONERA, Universit e Paris-Saclay, Palaiseau,  le-de-France, 91123, France
renato.murata@onera.fr louis.thioulose@onera.fr julien.marzat@onera.fr helene.piet-lahanier@onera.fr

² CNES, Sous-Direction Techniques Syst emes de Transport Spatial, Paris,  le-de-France, 75612, France
marco.galeotta@cnes.fr francois.farago@cnes.fr

ABSTRACT

A possible way to increase the reliability and availability of a system is to apply an Active Fault Tolerant Control (AFTC) algorithm. This thesis aims to use this algorithm in a multi-engine propulsive cluster with sensor and actuator faults. First, a Health Monitoring System (HMS) will be developed to monitor the entire propulsive cluster. The HMS will use model-based fault diagnosis techniques. Then, in case of actuator faults, the cluster will be reconfigured to minimize its effects. The reconfiguration can be made by using control allocation or modifying the control law of the engine. A simulation model of the entire cluster is under development. The model simulates the whole system, including the propellant feeding system, engines, and mechanical system. It will be used to study the effect of different faults on the system and compare different reconfiguration strategies.

Keywords: Control allocation, propellant feeding system, system of systems, fault tolerance, recovery.

1. PROBLEM STATEMENT

Reusable rockets are an innovation in the aerospace industry. Complete or partial recovery of rockets appears to be a promising way to reduce operating costs and environmental impacts. The next generation of European launchers is being designed with a multi-engine propulsive cluster composed of multiple Liquid-Propellant Rocket Engines (LPRE), a Thrust Vector Control (TVC), and a propellant feeding system.

The multi-engine cluster offers more reliability and availability with respect to a unique engine (Colas et al., 2019). Ideally, even if a failure occurs in one engine, the mission can be completed thanks to the remaining healthy engines. How-

ever, several steps should be addressed to achieve such a goal. First, the system should be able to detect, isolate, and identify all faults that might significantly impact the multi-engine cluster. Then, the system has to be reconfigured to mitigate the fault effect.

To solve the problems cited above, this thesis aims to develop an Active Fault Tolerant Control (AFTC) (Castaldi, Mimmo, & Simani, 2016) algorithm capable of treating actuators and sensor faults. This strategy will be based on the signals produced by the sensors. It must identify outliers from data and be able to operate in real-time under actuator faults.

A possible functional architecture of the multi-engine cluster is illustrated in Figure 1. Each motor has its own control law and Health Monitoring System (HMS) at the engine level. The role of the global HMS at the bay level is to monitor the health state of shared resources (TVC and propellant feeding system). Combining information from both levels of HMS, the overall state of the propulsive cluster can be established. The control allocation module's main objective is to translate the reference coming from the trajectory management module into feasible instructions to each engine, taking into account the health state of the cluster.

Concretely, the development of an HMS and an allocation module is linked to numerous open issues:

- The fault detection can be made at the individual engine level via an HMS filter (Sarotte, Marzat, Piet-Lahanier, Ordonneau, & Galeotta, 2020). On the other hand, the faults could be detected using information from the shared resources, at the bay level.
- The system reconfiguration under actuators fault can be made at the bay level, using an optimal control allocation (Johansen & Fossen, 2013) (Abauzit & Marzat, 2013) strategy in the engine cluster. The reconfiguration can also be performed at the engine level, modifying its con-

Renato MURATA et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

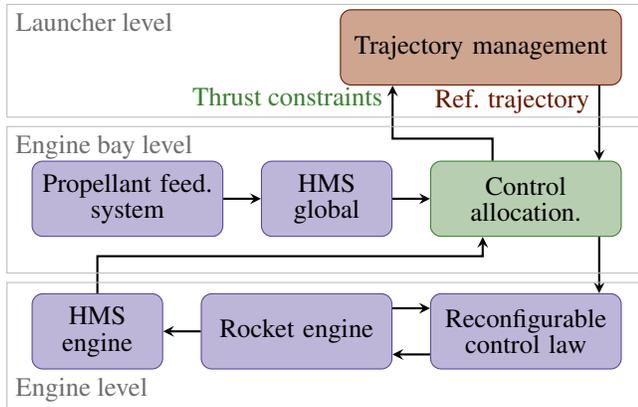


Figure 1. Possible functional architecture.

trol law.

- The consolidation of the sensor measurements is envisaged using analytical redundancy. The cluster can be seen as a system of systems: each engine shares its resources and capabilities, creating a highly interconnected system. This interdependency between systems can be used in favor of sensor measurement consolidation.
- If one actuator fault occurs, two different modules could make the trajectory reconfiguration. Either the trajectory module recalculates a feasible reference considering the degraded state, or the control allocation module generates its own optimal reference.

2. STATE OF THE ART

Fault Detection and Diagnosis (FDD) techniques in rocket engines have been studied since the 1970s, and it is a vital part of an HMS. In the early years, the health monitoring of rocket engines was made by observing some important operational parameters with fixed redlines values (Wu, 2005). In the literature, the Space Shuttle program carried out by NASA is an important source of information concerning HMS in LPRE. In (Hawman, Galinaitis, Tulpule, & Mattedi, 1990), different fault detection techniques are tested, and an HMS is proposed. The HMS has one output: yes/no to shut down the engine. According to the report, about 900 Space Shuttle Main Engine (SSME) failure modes can be identified with the database available at that time. The HMS focused on a small group of faults that directly impact the engine safety, but at the same time, they can be detected and treated. The minimization of the impact of those faults was the target of the HMS. Data-driven algorithms such as pattern recognition and Autoregressive Moving Average (ARMA) models were tested for fault detection. Another data-driven method, using bivariate time-series analysis, has been implemented in (Tsutsumi et al., 2021). The algorithm is applied to the Reusable Sounding Rocket (RSR) was developed by the Japan Aerospace Exploration Agency (JAXA). In (Iannetti, Marzat,

Piet-Lahanier, Ordonneau, & Vingert, 2015), model-based methods for fault diagnosis were tested in a rocket engine demonstrator. Under those circumstances, parameter identification and Kalman filter were applied. The modeling of LPREs can be made by describing the thermo-fluid and mechanical dynamics of the engine. Those models can be complex to simulate, requiring specific platforms. In (Pérez-Roca et al., 2019), an overview of modeling techniques of LPREs for control applications is given. The linearized version of the nonlinear thermodynamic model is often used to generate control laws. Control allocation algorithms can be used to control over-actuated mechanical systems. This algorithm can also be applied for the Fault Tolerant Control (FTC). In (Marks, Whidborne, & Yamamoto, 2012) an allocation scheme is used for FTC of an eight-rotor Unmanned Air Vehicle (UAV). If the UAV is exposed to rotor failures, the control allocation scheme maintains its stability and performance.

3. EXPECTED CONTRIBUTION

There is very few open literature on the HMS in a multi-engine propulsive cluster. Despite the fact that the HMS implemented in the Space Shuttle is widely documented, the system was developed for each engine separately. The multi-engine characteristic of the cluster was not taken into account. An HMS using the measurements from the shared resources of the propulsive cluster has not been investigated yet. This thesis aims to investigate different solutions considering the specificity of the multi-engine propulsive cluster from an HMS and control reconfiguration point of view. In particular, the fault diagnosis techniques applied to shared resources of the cluster, like reservoirs and propellant feeding lines. In addition, we intend to compare different reconfiguration methods in case of actuator fault.

4. RESEARCH PLAN

The associated research plan can be divided into three main tasks:

1. Modeling and simulation: a first mandatory task is to build a representative model of the system studied. It should be composed of LPREs, TVC, and a propellant feeding system with reservoirs and feeding lines. Then, the chosen failure modes can be simulated.
2. Fault diagnosis: in this phase, we intend to use model-based algorithms for fault detection, isolation, and identification. Model-based algorithms were chosen because it does not rely on recorded data. Reusable launchers are relatively new. Therefore flight data are scarce, especially data recorded under faulty conditions.
3. System reconfiguration: As discussed before, the system reconfiguration can be made at the bay level, using control allocation techniques, or at the engine level. The comparison of those strategies will be part of the study.

4.1. Work performed and remaining work

The modeling task has been the main focus of the work carried out so far. A rocket with three engines is used as a reference to our model. The rocket is considered to be powered by three 1000kN class engines, which will use liquid oxygen and liquid methane propellants. The CNES developed one OD model of the engine on the simulation software CARINS. The models of two subsystems have been derived: the plant dynamics and the propellant feeding lines. The plant dynamics inputs are the force and the position of each engine, and its output is the resulting force of the launcher. The propellant feeding lines are the system that connects the engines with the reservoirs. Its inputs are the inlet temperature and pressure of the propellant and the outlet mass flow. The outputs are the outlet temperature and pressure. A simplified diagram of the main models of the cluster is illustrated in figure 2.

We developed the propellant feeding system on Simulink® using the Simscape™ library. We simulated the model of the engine in nominal and faulty scenarios. Then, we recorded the evolution of the inlet mass flow of the engine. The mass flow is used as input to our feeding system, and we can study the impact of the simulated scenarios on the feeding system. The next step is to connect and validate the three existing models to the same platform. Then, steps 2 and 3 of the research plan will be studied.

5. CONCLUSION

Active fault-tolerant control is a promising way to increase the reliability and availability of a engine propulsive cluster. Mainly due to the hardware redundancy characteristics of the cluster. Different strategies to the reconfiguration problem can be considered, namely at the individual engine level or at the launcher level, and this thesis aims to implement and compare them. The steps that must be followed to study this problem were formulated. From the beginning, most of the effort is concentrated on building a representative model of the multi-engine cluster. Then after analyzing the subsystem models and the available measurements, dedicated methods will be selected and evaluated. It is foreseen that techniques relying on physics-based models would be better suited given the characteristics of the problems formulated in this paper.

REFERENCES

Abauzit, A., & Marzat, J. (2013). A multiple-observer scheme for fault detection, isolation and recovery of satellite thrusters. In *Advances in aerospace guidance, navigation and control* (pp. 199–214). Springer.

Castaldi, P., Mimmo, N., & Simani, S. (2016). Fault diagnosis and fault tolerant control strategies for aerospace systems. In *Proceedings of the 3rd conference on control and fault-tolerant systems* (pp. 684–689). Barcelona, Spain.

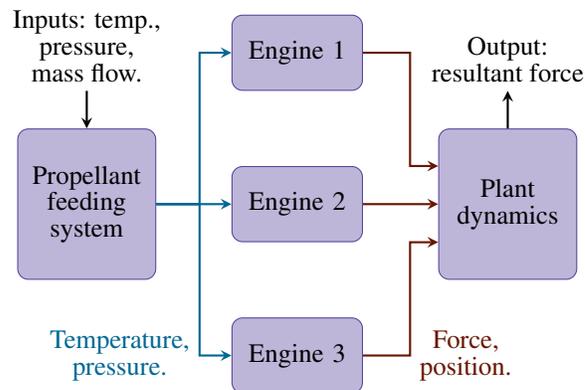


Figure 2. Simplified diagram of the multi-engine propulsive cluster model.

Colas, S., Gonidec, S., Saunois, P., Ganet, M., Remy, A., & Leboeuf, V. (2019). A point of view about the control of a reusable engine cluster. In *Proceedings of the 8th European conference for aeronautics and space sciences*. Madrid, Spain.

Hawman, M. W., Galinaitis, W. S., Tulpule, S., & Mattedi, A. K. (1990). Framework for a space shuttle main engine health monitoring system. *Report 185224, NASA, USA*.

Iannetti, A., Marzat, J., Piet-Lahanier, H., Ordonneau, G., & Vingert, L. (2015). Fault diagnosis benchmark for a rocket engine demonstrator. *IFAC-PapersOnLine*, 48(21), 895–900.

Johansen, T. A., & Fossen, T. I. (2013). Control allocation—a survey. *Automatica*, 49(5), 1087–1103.

Marks, A., Whidborne, J. F., & Yamamoto, I. (2012). Control allocation for fault tolerant control of a vtol octorotor. In *Proceedings of 2012 UKACC international conference on control* (pp. 357–362). Cardiff, United Kingdom.

Pérez-Roca, S., Marzat, J., Piet-Lahanier, H., Langlois, N., Farago, F., Galeotta, M., & Le Gonidec, S. (2019). A survey of automatic control methods for liquid-propellant rocket engines. *Progress in Aerospace Sciences*, 107, 63–84.

Sarotte, C., Marzat, J., Piet-Lahanier, H., Ordonneau, G., & Galeotta, M. (2020). Model-based active fault-tolerant control for a cryogenic combustion test bench. *Acta Astronautica*, 177, 457–477.

Tsutsumi, S., Hirabayashi, M., Sato, D., Kawatsu, K., Sato, M., Kimura, T., & Hashimoto, T., & Abe M. (2021). Data-driven fault detection in a reusable rocket engine using bivariate time-series analysis. *Acta Astronautica*, 179, 685–694.

Wu, J. (2005). Liquid-propellant rocket engines health-monitoring—a survey. *Acta Astronautica*, 56(3), 347–356.

Artificial-intelligence-based maintenance scheduling for complex systems with multiple dependencies

Van-Thai Nguyen, Phuc Do, and Alexandre Voisin

Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France

van-thai.nguyen@univ-lorraine.fr

phuc.do@univ-lorraine.fr

alexandre.voisin@univ-lorraine.fr

ABSTRACT

Maintenance planning for complex systems has still been a challenging problem. Firstly, integrating multiple dependency types into maintenance models makes them more realistic, however, more complicated to solve and analyze. Secondly, the number of maintenance decision variables needed to be optimized increases rapidly in the number of components, causing computational expensive for optimization algorithms. To face these issues, this thesis aims to incorporate multiple kinds of dependencies into maintenance models as well as to take advantage of recent advances in artificial intelligence field to effectively optimize maintenance polices for large-scale multi-component systems.

1. MOTIVATION AND RESEARCH PROBLEM STATEMENT

Due to higher demand in performance and safety, modern engineering systems nowadays are often composed of many components, where different inter-component dependencies can exist (Keizer, Flapper, & Teunter, 2017). Omitting component dependencies in maintenance modeling could result in high maintenance cost and suboptimal maintenance plan. Therefore, it is necessary to integrate them into maintenance models.

Maintenance policies can be classified into corrective (CM) and preventive (PM) strategy. CM carries out maintenance actions on failed machines, which is usually associated with high related costs due to unexpected production losses as well as unscheduled maintenance costs. In contrary, PM aims at maintaining functioning machines to prevent sudden failures, hence, to reduce downtime costs. PM interventions can be planned either in time-oriented or condition-based manner (CBM). However, the later appears to be more advantageous. Particularly, it allows to proactively make maintenance de-

isions based on degradation states of maintained machines instead of on a fixed calendar. Moreover, recent advances in sensing and information technology allow rich degradation data to be collected enabling CBM to become a popular and sophisticated approach for maintenance decision-making and optimization.

Whereas CBM optimization processes might be effectively achieved for single-unit systems due to the small number of decision variables needed to be optimized, the ones for multi-component systems suffer from the curse of dimensionality. Specifically, the number of decision variables grows rapidly as the number of components increases, causing computational expensive for optimization algorithms (Zhang & Si, 2020). Fortunately, recent advances in the field of artificial intelligence (AI) open a new direction to solve large maintenance decision-making problems. Therefore, how to take advantage of these advances to effectively plan maintenance actions for complex systems is a crucial issue.

Based on the above analysis, the objective of this thesis is to leverage AI techniques to optimally schedule maintenance operations for large-scale multi-component systems taking into account the impact of component dependencies.

2. THE STATE-OF-THE-ART

2.1. Component dependencies

Component dependencies fall into three primary groups which are economic, structure and stochastic dependence (Nicolai & Dekker, 2008). The economic dependence means that the joint cost of maintaining a group of several components is not equal to the cost of maintaining them separately. The structural dependence implies that repairing one component requires at least dismantling or maintaining other units.

Component stochastic dependence might be triggered by the failure or degradation levels of one component (failure- and degradation-based stochastic dependence). The number of studies investigating the later is small in comparison to the

Van-Thai Nguyen et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

one studying the former. Several of them are reviewed in the following. The stochastic dependence characterized by state-state interactions meaning that the degradation evolution of a component is a function of its own state as well as other components' state, is studied in (Rasmekomen & Parlikad, 2014) for an industrial cool box in which component degradation is modeled using Gaussian process. The state-rate interaction, which implies that degradation levels of a component accelerate the degradation process of its dependent components, is investigated in (Bian & Gebraeel, 2014) for a networked system using a system of continuous stochastic differential equations, and in (Do, Assaf, Scarf, & Iung, 2019) for a gearbox system whose component degradation processes are described by Gamma processes. More recently, (Wang, Li, Chen, & Liu, 2022) investigated the impact of the degradation acceleration of one component on the degradation rate of other components (rate-rate interactions) in a continuous way over time for general multi-component systems using Wiener process. It can be noticed that the aforementioned articles describe the degradation-based stochastic dependence based on continuous stochastic processes. *Hence, there is still a gap in modeling such kind of stochastic dependence using discrete stochastic processes.*

Moreover, almost all existing maintenance models consider only one specific kind of dependency since integrating more than one makes them more complicated to analyze as well as more difficult to solve (Keizer et al., 2017). However, multiple dependency types in practice exist in many systems. For example, a gearbox system is studied in (Do et al., 2019), which suffers from both economic and stochastic dependence. *Therefore, incorporating multiple kinds of dependencies into maintenance models is necessarily taken into account.*

2.2. Maintenance optimization

CBM policies can be divided into two main groups: direct mapping and threshold-based policy. While the former maps directly from component degradation measurements to maintenance actions, the later first compares component states to predefined thresholds, and then choose maintenance actions accordingly. As mentioned previously, CBM optimization for multi-component systems suffers from the curse of dimensionality that causes computation expensive for optimization algorithms.

Recent advancements in the field of reinforcement learning (RL) give rise to direct mapping approaches by providing new tools to deal with maintenance decision optimization for large-scale systems. Particularly, deep RL algorithms (DRL) are employed to minimize maintenance costs for systems with extremely large state spaces showing better performance in comparison to threshold-based policies (Huang, Chang, & Arinez, 2020). However, DRL algorithms belong to the class of single-agent RL algorithms that is shown in literature to

suffer from the problem of large action spaces. Fortunately, the framework of multi-agent DRL (MADRL) appears as a promising solution to this challenge, which has been received recently increasing attention from maintenance researchers (Huang et al., 2020; Andriotis & Papakonstantinou, 2021). *Consequently, how to take advantage of MADRL algorithms to effectively optimize maintenance decisions of large-scale multi-component systems is a crucial issue.*

Furthermore, it is vital in maintenance decision optimization to correctly define the objective function that often requires the system maintenance cost model. From a practical point of view, repair actions are usually grouped in each maintenance intervention due to the economic dependence between maintained components, which leads to the fact that individual costs, such as setup cost, spare part cost, labor cost, costs of maintaining each component are not recorded separately, instead, only total cost is documented. As a result, the requirement of separately collecting individual maintenance costs to construct the cost model at system level in almost all existing maintenance decision optimization algorithms used for multi-component systems is less practical. *Therefore, it raises the need for constructing a system cost predictor that can get rid of the demand of accessing individual maintenance costs.*

3. EXPECTED CONTRIBUTIONS

To support the scientific issues analyzed in the state-of-the-art section, the expected contributions are identified as follows:

1. Take advantage of AI techniques to solve practical maintenance decision-making problems.
2. Propose maintenance models taking into consideration multiple dependency types.
3. Propose approaches for modeling the degradation-based stochastic dependence using discrete stochastic processes.
4. Develop and/or improve MADRL algorithms to optimize effectively maintenance decisions of large-scale multi-component systems.

4. ACHIEVED WORKS

An AI-based framework is proposed in (Nguyen, Do, Voisin, & Iung, 2021) for maintenance decision-making in the case of unknown cost model at system level (*expected contribution 1*).

The proposed framework consists of two main stages which are offline training and online decision-making as depicted in figure 1. The former aims at optimizing maintenance policies based on collected data, and the later involves realizing optimized policies. Moreover, the first stage is composed of two main phases. The first one aims to learn system cost model using artificial neural networks, and to estimate component degradation probability transition matrices. The objective of the second phase is to first construct learning envi-

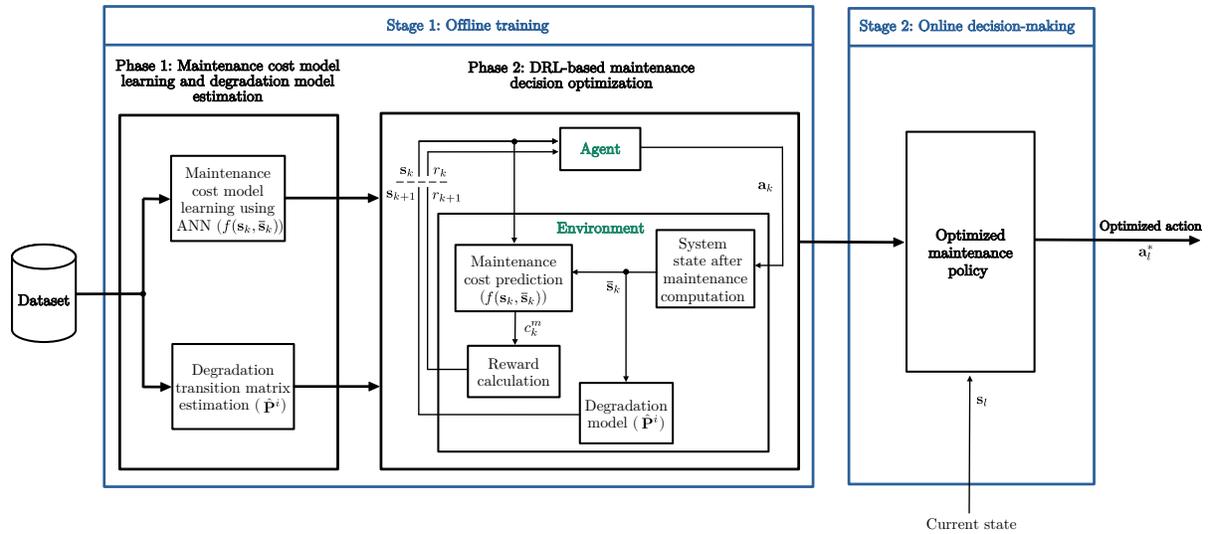


Figure 1. Illustration of AI-based maintenance framework for multi-component systems.

ronments dedicated to DRL algorithms employing the trained cost models and estimated transition matrices from the first phase, and then to train DRL agents to optimize maintenance decisions by letting them interact with the constructed environment. For more details about the framework, please take a visit to (Nguyen et al., 2021).

5. CURRENT WORKS

The current works focus on maintenance planning for multi-component systems suffering from economic and stochastic dependence (*expected contribution 2*). The economic dependence is described by two levels of setup cost sharing: system setup cost caused by administrative handling or transportation of spare parts, and component-type setup cost originated from the requirement of specific tools or repairman skills.

The stochastic dependence through state-rate interactions is modeled using a new approach based on the framework of Markov decision processes (*expected contribution 3*). Figure 2 illustrates the degradation interactions in a two-component system in which each component has four condition states. It can be noticed that when a component degrades to a more serve state, its dependent components tend to degrade faster since their probabilities of staying in current states decrease while their probabilities of transitioning to worse conditions increase.

In order to take the *fourth expected contribution* into account, we customize a MADRL algorithm, namely WQMIX (Rashid, Farquhar, Peng, & Whiteson, 2020), in the case where system states can be fully observed to obtain cost-effective policies. Indeed, the algorithm takes advantage of the branching dueling network architecture (Tavakoli, Pardo, & Kormushev, 2018) to allow achieving linear increase in the size of the

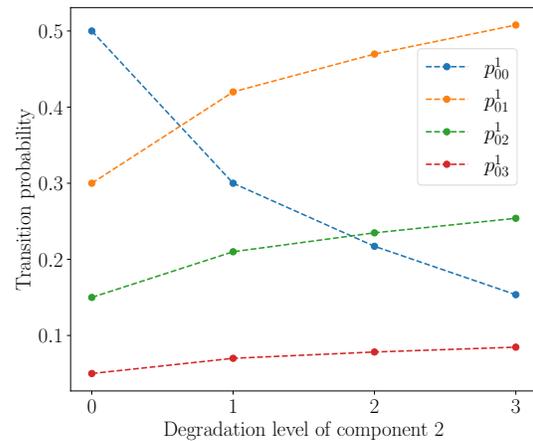


Figure 2. Changes in transition probabilities of component 1

output layer of deep Q-networks when the number of system components grows and the monotonic decomposition scheme for joint action-value functions (Rashid et al., 2020) to enable maintenance decision-making consistency at component and system level.

A comparative experiment is conducted on a 5-component system to verify the performance of the customized algorithm as well as to investigate the impact of component dependencies on optimized policies. The experimental results depicted in figure 3 show that BDQ (a MADRL algorithm also uses the branching network) and Dueling DDQN (a DRL algorithm) are incapable of approaching the optimal policy obtained by the customized WQMIX and VI (a dynamic programming algorithm), however, the optimization time of the customized algorithm (2 hours) is much less than the one of VI (9 hours). Moreover, it can be noticed that the optimal

policy has an incentive to perform maintenance actions frequently and in groups due to the impact of component economic and stochastic dependence.

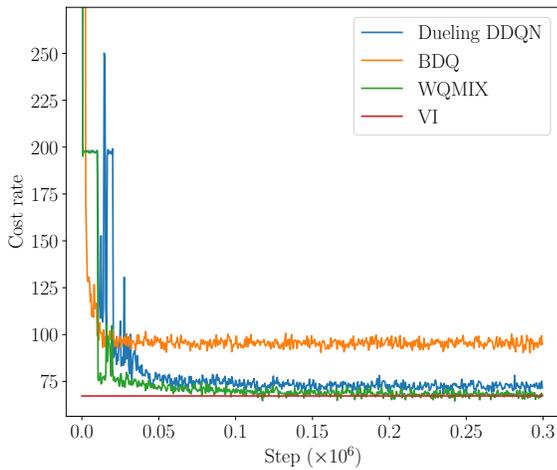


Figure 3. Evolution of cost rates during training

6. CONCLUSIONS

In this paper, the thesis’s objective is identified, which is to take advantage of AI techniques to optimally schedule maintenance for large-scale multi-component systems taking into account the impact of component dependencies. To support this research direction, an AI-based framework is proposed in (Nguyen et al., 2021) to tackle maintenance decision-making in the case of unknown system cost models. The current works focus on modeling maintained systems with discrete-state components which suffer from stochastic and economic dependence as well as on customizing MADRL algorithms to effectively optimize maintenance decisions of large-scale systems.

The thesis’s future work will focus on modeling methods that can integrate all three dependency types into maintenance models. Improving the learning speed of MADRL algorithms will also be considered.

ACKNOWLEDGMENT

This thesis is part of the AI-PROFICIENT project which has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No 957391.

REFERENCES

Andriotis, C., & Papakonstantinou, K. (2021). Deep re-

inforcement learning driven inspection and maintenance planning under incomplete information and constraints. *Reliability Engineering & System Safety*, 212, 107551.

Bian, L., & Gebraeel, N. (2014). Stochastic framework for partially degradation systems with continuous component degradation-rate-interactions. *Naval Research Logistics (NRL)*, 61(4), 286–303.

Do, P., Assaf, R., Scarf, P., & Iung, B. (2019). Modelling and application of condition-based maintenance for a two-component system with stochastic and economic dependencies. *Reliability Engineering & System Safety*, 182, 86–97.

Huang, J., Chang, Q., & Arinez, J. (2020). Deep reinforcement learning based preventive maintenance policy for serial production lines. *Expert Systems with Applications*, 160, 113701.

Keizer, M. C. O., Flapper, S. D. P., & Teunter, R. H. (2017). Condition-based maintenance policies for systems with multiple dependent components: A review. *European Journal of Operational Research*, 261(2), 405–420.

Nguyen, V. T., Do, P., Voisin, A., & Iung, B. (2021). Reinforcement learning for maintenance decision-making of multi-state component systems with imperfect maintenance. In *Proceedings of the 31st european safety and reliability conference*.

Nicolai, R. P., & Dekker, R. (2008). Optimal maintenance of multi-component systems: a review. *Complex system maintenance handbook*, 263–286.

Rashid, T., Farquhar, G., Peng, B., & Whiteson, S. (2020). Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*.

Rasmekomen, N., & Parlikad, A. K. (2014). Optimising maintenance of multi-component systems with degradation interactions. *IFAC Proceedings Volumes*, 47(3), 7098–7103.

Tavakoli, A., Pardo, F., & Kormushev, P. (2018). Action branching architectures for deep reinforcement learning. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 32).

Wang, Y., Li, X., Chen, J., & Liu, Y. (2022). A condition-based maintenance policy for multi-component systems subject to stochastic and economic dependencies. *Reliability Engineering & System Safety*, 219, 108174.

Zhang, N., & Si, W. (2020). Deep reinforcement learning for condition-based maintenance planning of multi-component systems under dependent competing risks. *Reliability Engineering & System Safety*, 203, 107094.

Contribution to the design and implementation of a reflexive cyber-physical system: application to air quality prediction in the vallées des gaves

Sylvain Poupry, Cédric Béler, and Kamal Medjaher

Laboratoire Génie de Production, ENIT - Toulouse INP, 47 Avenue Azereix, Tarbes, 65000, France

sylvain.poupry@enit.fr

cedrick.beler@enit.fr

kamal.medjaher@enit.fr

ABSTRACT

This thesis aims to set up a scientific approach to monitor and take preventive actions on the air quality for the actors of a territory not covered by conventional measuring stations. Thus, a Cyber-Physical System (CPS) approach combined with Prognostics Health Management (PHM) methodologies is chosen to move toward a self-monitoring and self-reconfiguration system. To collect data in an inexpensive manner, measurement stations with low-cost sensors (LCS) are developed. LCS have drawbacks and the first part of this thesis is the use of redundancy and a proposed algorithm to increase their hardware and data reliability. A first station is deployed as proof of concept and the region is already receiving real-time data. The next phase is to build forecasting models to help authorities make decisions.

Keyword: Cyber-Physical System, Internet of Things, Prognostics & Health Management, air quality, citizen engagement, natural system.

1. MOTIVATION AND RESEARCH PROBLEM STATEMENT

The objective of the thesis is the proposal and implementation of an innovative scientific approach for the monitoring and prediction of air quality on the territory of the Communauté de Commune Pyrénées Vallées des Gaves (CCPVG). Located in the South of France, in the Occitanie region, in the South of the Hautes-Pyrénées department, the CCPVG is an inter-municipal territory of about 1000 km² and covered by forests and semi-natural environments as well as high mountains.

The territory of the CCPVG is confronted with peaks of air pollution linked to wood heating, firewood burning, and road traffic. These emissions of specific pollutants hide more se-

rious risks such as background pollution, which is more dangerous for health because it acts over the long term. This is why it is important to monitor pollution throughout the year. In France, air quality monitoring is delegated to the Air Quality Associations (ASQA) under the supervision of ATMO France. In the case of the CCPVG, ATMO Occitanie is in charge of air quality monitoring and provides three tools: a real-time and continuous measurement network with conventional stations, pollution maps and daily predictions, and an annual regional emission inventory.

However, these three tools are insufficient for many territories similar to the GVCC. First, the regional inventory is an annual and global synthesis of pollution emissions and their impact on air quality. This inventory is published after a considerable time of analysis in addition to the time of data collection. Moreover, it is an estimation based on statistical data and does not allow to have a "real-time" vision of the situation and to act quickly to limit the pollution effects. Secondly, the pollution maps and predictions are based on measurements made by the fixed stations of ATMO Occitanie and there are only two fixed measuring stations in the Hautes-Pyrénées department. As there is no fixed station on the territory of the CCPVG, the data provided by ATMO Occitanie are extrapolations collected on a reduced number of sensors which is not necessarily optimal. Indeed, this territory presents a particular topology (mountains and valleys) that generates local phenomena of pollution concentration. These local pollutions are detected by the inhabitants but escape the monitoring of the ASQA.

This thesis aims at setting up a rigorous approach that will make it possible to provide elements of scientific, tangible, and quantified knowledge to the citizens and the actors of the territory in charge of monitoring and managing the air quality. Based on these elements, the latter will be able to better apprehend the pollution problem, estimate, predict and evaluate its severity, and define preventive action plans (sensitiza-

Sylvain Poupry et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

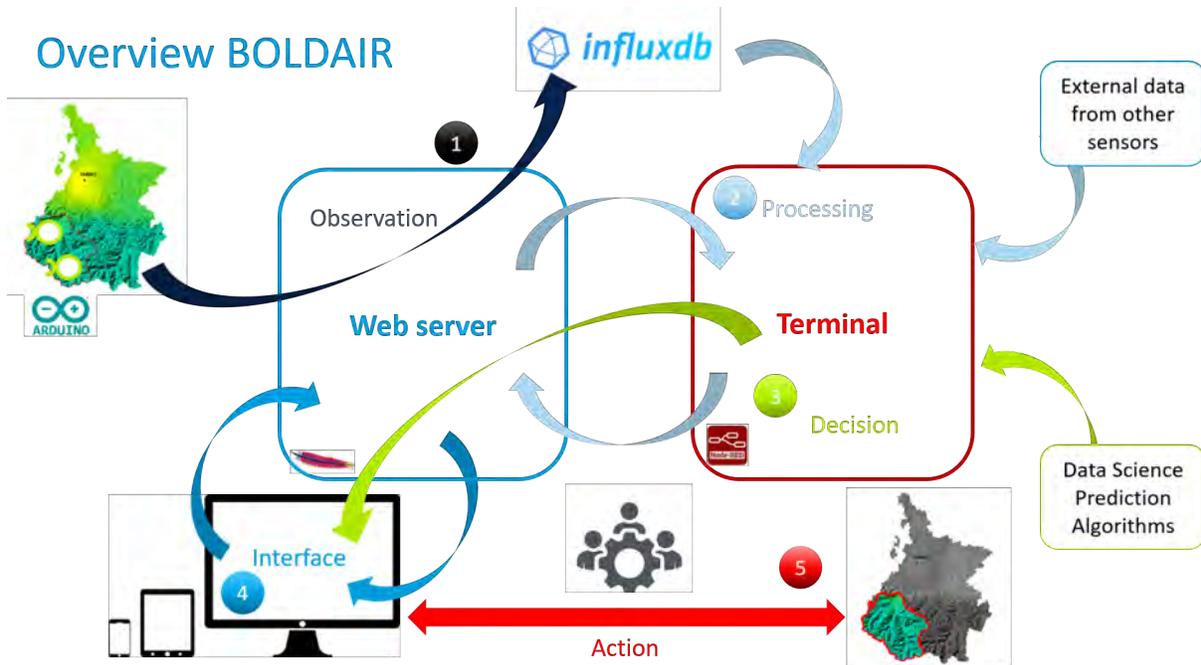


Figure 1. The five modules of the CPS

tion of the inhabitants, incitement to the use of heating fewer pollutants emissions, incitement of the stockbreeders to less resort to ecoburials, etc.), but also and especially pro-active (regulation of the road traffic and industrial pollution by anticipation of future degradation of the air quality). The proposed approach will take the form of a Cyber-Physical System (CPS) equipped with observation, processing, decision, action, and communication capabilities. For that, three axes of scientific developments will be approached:

1. Monitoring dimension: instrumentation of the territory to have a *real-time* vision of the air quality;
2. Prediction dimension: the anticipation of air quality degradation and proposal of proactive actions;
3. Societal dimension: involvement of various actors (elected officials, inhabitants, tourists, etc.).

2. A BRIEF DISCUSSION ON THE STATE-OF-THE-ART

Pillars of Industry 4.0, the CPS appeared in 2006. Heirs to the Internet of Things, they have become a priority focus. There are several definitions of CPS depending on the domain where they are used (Monostori, 2014). In this thesis, a CPS is defined as "an orchestration of computers and physical systems. Embedded computers monitor and control physical processes, usually with feedback loops, where physical processes affect computations and vice versa" (Lee, Bagheri, & Kao, 2015). CPS form a closed loop between the cyber/digital and physical worlds based on state detection, real-time analysis, scientific decision-making, and precise ex-

ecution. Then, different CPS architecture are proposed in literature (Monostori, 2014; Lee et al., 2015; Shi, Wan, Yan, & Suo, 2011), and the one we will consider is the "5C" architecture proposed by Lee in 2015 (Lee et al., 2015). This article is a guide to incrementally building a high-value CPS. The first step in building a CPS is augmenting the physical system by adding computational, communication, and analytical capabilities, i.e., the territory to be observed will be augmented with a network for measuring pollution concentrations. For territories not covered by measuring stations, Low Cost Sensor (LCS) networks are an opportunity to monitor Air Quality (Castell et al., 2017). Less expensive, in the order of x10 to x100, they are easier to deploy and require few qualified personnel. Although they provide coarse measurements, their strength lies in their spatial dimension, which allows the detection of local pollution (Kumar et al., 2015). However, they have several drawbacks, notably in terms of repeatability, reliability, and lifetime (Morawska et al., 2018). Also, research works are insufficient about LCS deployment experimental duration. Indeed, the duration of the published studies varies from a few days to 4 months on average with an exception made for a study carried out over 11 months (Bauerová, Šindelářová, Rychlík, Novák, & Keder, 2020). These durations are thus insufficient especially when the study subject must be monitored continuously over several years.

3. NOVELTY AND SIGNIFICANCE RELATIVE TO THE STATE OF THE ART

Based on the 5C architecture, the analysis of the state of the art shows that, in practice, the total integration of functionalities within a CPS is only very rarely achieved. The goal is then to achieve a complete CPS capable of observing, processing, analyzing, predicting, acting, and communicating, but also of self-monitoring to detect any anomalies the CPS may experience and to self-reconfigure it accordingly. We get then of a *reflexive* CPS, that is to say introspective, having consciousness of itself (observe, model, and evaluate) and capable of acting on itself (switching in degraded mode, adaptation, evolution). For this purpose, we propose to integrate PHM (Prognostics and Health Management) functionalities in the architecture to assess online the health status of the CPS (Medjaher, Zerhouni, & Gouriveau, 2016; Atamuradov, Medjaher, Dersin, Lamoureux, & Zerhouni, 2017). This integration will provide greater confidence in the adaptability of the prediction system based on networked components (sensors, processing, and actuators).

At the level of the physical system augmentation, which is the first step of the CPS construction, the pollutant concentrations are measured with a measurement network deployment. However, the deployments from the literature do not take into account the relative reliability of the sensors and the possible improvements found in the sensor technologies (clustering, post-processing with Machine Learning, etc.). Novelty of the work proposed in this thesis lies in the use of modular redundancy at the measurement point. Indeed, the fact of measuring the same parameter several times at the same place with several sensors makes it possible to group the LCS to increase their reliability but also to compare them to detect their possible failures. The interest of this approach is to go further at the level of the measurement horizon and to avoid the drawbacks that are not solved during standard deployments with LCS.

4. APPROACH AND PROPOSED EXPERIMENTS

The first step consists in proposing an elementary system as modular CPS (figure 1). The CPS is composed of five linked modules (Observation, Processing, Decision, Interface, and Action) aiming to provide services.

The second step is to design and develop the first Observation module by selecting LCS and making them smart. The combination of LCS with a development board having communication and calculation capabilities allows to build a Smart Sensor (SmS). Its purpose is to measure parameters at a point, to calculate their concentration and to transmit a vector of data which is the concatenation of the processed measures with the name of the associated parameters.

The third approach is to propose an architecture composed of

measuring stations that exploit SmS combined with an Aggregator. The latter element receives the vectors of data emitted from the three SmS arranged in the same measurement perimeter and measuring the same number of parameters. Then, it restructures the data and aggregates them to perform a synthesis. The next step will be to process the data in order to make this architecture reliable and resilient for air quality monitoring (Processing module in the CPS).

The final step is making the Interface and Decision modules to propose a decision-support for authorities to act on the air quality and the CPS itself.

5. WORK IN PROGRESS OR RESULTS

The design of the SmS has been completed and they have been deployed. There are currently three of them. The acquired data are sent to a central server with a graphical interface that allows tracking the pollution evolution in real-time. This central server currently integrates the Aggregator which could have its own calculation unit depending on the calculation needs and the following deployments.

The work in progress is at the level of the functionalities of the Aggregator. The first function, as a reminder, is to receive data from the three (or more) SmS, store, and restructure them by parameters. The second function is the detection of abnormal raw data due to sensors faults. The work in progress consists in creating detection algorithms at the sensor level and at the Air Quality level. In a first step, the detection of sensor anomalies allows making the data more reliable and replacing the faulting LCS to restore the confidence in the data of the measuring station. Then, at the air quality level, other algorithms will allow to follow the air quality and to detect pollution peaks. The evolution of these abnormal concentrations over time will allow predicting the next pollution peaks. This work will help the competent authorities to take decisions and to follow the effectiveness of their actions to improve the air quality.

6. DISCUSSION OF APPLICATIONS AND THE CONTRIBUTIONS OF THE WORK

The deployment of the first measurement station validates the proof of concept of the observation part of the CPS. Other stations will be deployed and the territory will be "augmented" for the measurement of air quality. Indeed, these stations offer syntheses on two aspects: The air quality as an observed system and the CPS itself by monitoring its components. The implementation of the PHM at the hardware level and the actions to replace failing sensors add reflexivity to the CPS. This feature allows (with human intervention) an extensive horizon of measurements and reliability to the data for the final objective of monitoring and forecasting air quality on the territory of the CCPVG. Once the CPS is fully implemented, it will provide a reliable decision support to local authorities to

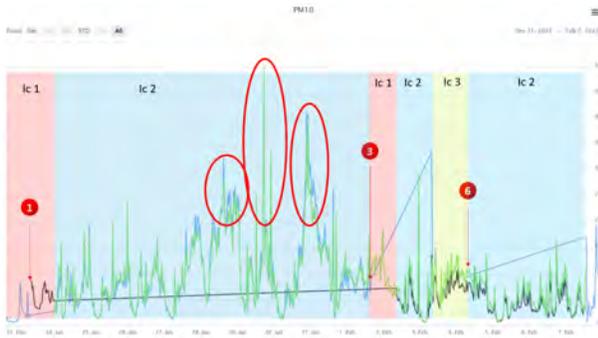


Figure 2. This is an example of output of the CPS.

act directly on pollution in case of alert and to observe on the long term the measures taken and their efficiency. It will also be able to provide a contribution to ATMO FRANCE and ATMO OCCITANIE by sharing data and thus to carry out data crossings to check if the pollution episodes detected by the association are also detected by the measurement network. These information will be exploited to specify the pollution maps and to refine the territory grid.

REFERENCES

- Atamuradov, V., Medjaher, K., Dersin, P., Lamoureux, B., & Zerhouni, N. (2017). Prognostics and Health Management for Maintenance Practitioners - Review, Implementation and Tools Evaluation. *International Journal of Prognostics and Health Management*, 8(3). (Number: 3) doi: 10.36001/ijphm.2017.v8i3.2667
- Bauerová, P., Šindelářová, A., Rychlík, S., Novák, Z., & Keder, J. (2020, May). Low-Cost Air Quality Sensors: One-Year Field Comparative Measurement of Different Gas Sensors and Particle Counters with Reference Monitors at Tušimice Observatory. *Atmosphere*, 11(5), 492. (Number: 5 Publisher: Multidisciplinary Digital Publishing Institute) doi: 10.3390/atmos11050492
- Castell, N., Dauge, F. R., Schneider, P., Vogt, M., Lerner, U., Fishbain, B., ... Bartonova, A. (2017, February). Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates? *Environment International*, 99, 293–302. doi: 10.1016/j.envint.2016.12.007
- Kumar, P., Morawska, L., Martani, C., Biskos, G., Neophytou, M., Di Sabatino, S., ... Britter, R. (2015, February). The rise of low-cost sensing for managing air pollution in cities. *Environment International*, 75, 199–205. doi: 10.1016/j.envint.2014.11.019
- Lee, J., Bagheri, B., & Kao, H.-A. (2015, January). A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3, 18–23. doi: 10.1016/j.mfglet.2014.12.001
- Medjaher, K., Zerhouni, N., & Gouriveau, R. (2016). *From Prognostics and Health Systems Management to Predictive Maintenance I: Monitoring and Prognostics*. John Wiley & Sons. (Google-Books-ID: usND-DQAAQBAJ)
- Monostori, L. (2014). Cyber-physical production systems: Roots, expectations and r&d challenges. *Procedia CIRP*, 17, 9-13. (Variety Management in Manufacturing) doi: https://doi.org/10.1016/j.procir.2014.03.115
- Morawska, L., Thai, P. K., Liu, X., Asumadu-Sakyi, A., Ayoko, G., Bartonova, A., ... Williams, R. (2018, July). Applications of low-cost sensing technologies for air quality monitoring and exposure assessment: How far have they gone? *Environment International*, 116, 286–299. doi: 10.1016/j.envint.2018.04.018
- Shi, J., Wan, J., Yan, H., & Suo, H. (2011, November). A survey of Cyber-Physical Systems. In *2011 International Conference on Wireless Communications and Signal Processing (WCSP)* (pp. 1–6). doi: 10.1109/WCSP.2011.6096958

Combining Knowledge and Deep Learning for Prognostics and Health Management

Maximilian-Peter Radtke¹, Jürgen Bock²

^{1,2} *Technische Hochschule Ingolstadt, Ingolstadt, Germany*

maximilian-peter.radtke@thi.de

juergen.bock@thi.de

ABSTRACT

In the recent past deep learning approaches have achieved remarkable results in the area of Prognostics and Health Management (PHM). These algorithms rely on large amounts of data, which is often not available, and produce outputs, which are hard to interpret. Before the broad success of deep learning machine faults were often classified using domain expert knowledge based on experience and physical models. In comparison, these approaches only require small amounts of data and produce highly interpretable results. On the downside, however, they struggle to predict unexpected patterns hidden in data. This research aims to combine knowledge and deep learning to increase accuracy, robustness and interpretability of current models.

1. MOTIVATION AND RESEARCH QUESTION

Production halls are becoming more automated, efficient and flexible through the increasingly widespread adaption of the Industrial Internet of Things (IIoT). This provides the opportunity of working with minimal inventory and an optimal amount of work in progress. To guarantee the smooth sequence of operations nonetheless, requirements towards the functionality and reliability of machines are increasing. Additionally, all manufacturing companies are under the constant pressure of reducing costs. Therefore, state-of-the-art maintenance strategies for the ensured flow of processes and cost reduction are indispensable (Pawellek, 2016).

The basis for modern maintenance strategies is the rapid diagnosis and prognosis of faults using current machinery conditions, which makes it possible to base maintenance decisions on the expected remaining useful life (RUL). This is called predictive maintenance. In comparison to traditional maintenance strategies this approach can reduce machine down times by 35 % - 45 % and increase production by 20 % - 25 % (Selcuk, 2017). Prognostics and Health Management (PHM)

Maximilian-Peter Radtke et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

provides the methods and techniques to analyze condition monitoring data and make predictive maintenance possible.

In the last years the advent of deep learning (DL) has shown remarkable results in diagnosing and predicting failures (Fink et al., 2020). Nevertheless, multiple obstacles remain if these algorithms are to be used in practice. One issue is the availability of correctly labeled data needed for training and the question on how models can work with limited or even non-existent fault data. Many DL models have strong performance on specific data but deteriorate when confronted with minor domain changes. How to make models more robust is therefore a question worth exploring. A known issue with DL is the black box nature of the models. How the results can be explained nevertheless and if potential root causes can be inferred from these explanations are further open questions.

An attempt to tackle these issues in current state-of-the-art data driven DL models is the combination with knowledge. Hereby knowledge, which is formalized within a knowledge base, can consist of both physical models of fault processes and semantic knowledge describing the underlying system and relations. The resulting hybrid model is able to provide answers to the posed questions.

Consequently the overarching proposed research question for the PhD is the following: How can we improve state-of-the-art DL models for machinery fault classification and RUL prediction with the help of knowledge? The resulting methodology will be used to answer more specific questions: How can the amount of required labeled data be decreased? How can the explainability of black box DL approaches be increased and used for root cause analysis? How can DL models be made more robust towards outliers and minor domain changes?

2. STATE OF THE ART

Multiple approaches for detecting and predicting machinery faults exist. These can be grouped into experience-based, data-driven and physics-based models (Liao & Köttig, 2014). To create hybrid models these methods are combined to form new approaches, which alleviate the disadvantages of pure

models. Especially inducing data-driven models with knowledge based on experience and physics has been successful. Recent work in the PHM domain has shown to increase explainability and enable root cause analysis by taking into account input from human experts (Steenwinckel et al., 2021), make models more robust and decrease the amount of data needed through the simulation of data based on expert knowledge (Wang, Taal, & Fink, 2021), and increase overall performance by extending the feature space (Chao, Kulkarni, Goebel, & Fink, 2022).

Even though recent work in the direction of combining knowledge with data is promising, a lot of untapped knowledge in the shape of physical models (e.g. for rotating machinery (Cubillo, Perinpanayagam, & Esperon-Miguez, 2016)) and symbolic models, such as manufacturing ontologies (Cao, Zanni-Merk, & Reich, 2018), remains. Especially formal logic and semantic knowledge have not yet been studied in detail for mixing with DL in the PHM domain and provide fertile ground for further research.

3. CONTRIBUTION

Two possible approaches towards combining PHM specific knowledge with deep learning will be explored.

3.1. Pretraining DL models based on knowledge

The first idea is to use knowledge based on physical and domain expertise as pretraining for a DL model. Through this approach one can mimic the process of human learning, where clear theoretical instructions are learned initially and are later fine tuned by practical experience. A first visualization of the possible process is given in Fig. 1. The idea aims to decrease the amount of necessary labeled data and make the optimal weight configuration of the resulting neural network more robust towards data anomalies and minor domain changes. For the correct implementation multiple questions need to be answered: What is the appropriate formalism for the knowledge base? Does an existing formalism have to be extended for this use case? How can the knowledge base be used for pretraining? Should data simulated with help of the knowledge base or observational data be used for teaching knowledge to the neural network? Can the intrinsic explainability of the knowledge base be maintained after fine tuning?

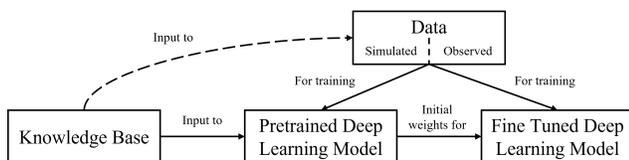


Figure 1. Potential high level process for pretraining with a knowledge base.

3.2. Splitting and combining DL tasks with knowledge

The second idea is to split the complex problem of failure diagnosis and prognosis into multiple sub-problems, which are solved via DL and are later combined to answer the initial question. Both the split into sub-problems and the combination of the outputs is done on the basis of underlying knowledge about the issue. For example, the occurrence of single known failure root causes can be predicted separately and later combined for possible fault inference. Or, different groups of features are defined, which are used to train different models independently, for which the combined output gives the final fault diagnosis. The proposed process is visualized in Fig. 2. By dividing the main problem into multiple sub-problems the amount of data needed is expected to decrease owing to the passing of explicit knowledge, which in turn does not need to be learned. Additionally, increased explainability due to the modular nature of the approach is achieved. Again, multiple questions arise when realizing the idea. How can the knowledge base be formalized? Does an existing formalism have to be extended for this use case? How can sub-questions be identified and inferred? How can the combination of different DL outputs be achieved?

Starting off, these two ideas will be examined in parallel and subsequently reflected on whether or not they should both be continued. In a later stage of research integrating the two streams is a possibility worth investigating.

In comparison to work in Sec. 2, where mainly physical knowledge was used to improve DL, the focus will be on additionally integrating symbolic knowledge for the union of different models and formalizing the knowledge base (both symbolic and physical) for wider reuse.

4. WORK IN PROGRESS

In keeping with the first proposed idea, using knowledge for training neural networks in the domain of rolling element bearing fault detection is being examined in detail as an initial step. In the current (work-in-progress) paper a knowledge base for fault classification is created by deriving expected physical attributes of different faults through vibration signals. This knowledge is used to create a similarity function for comparing input signals to expected faulty signals. Afterwards the similarity measure is incorporated into a DL model using a Logic Tensor Network (LTN) (Badreddine, Garcez, Serafini, & Spranger, 2022). This enables logical reasoning in the loss function, in which the decision process of an expert analyzing the input data is to be imitated. The combination of the symbolic loss function and the underlying knowledge base enables better explainability of the classification results and achieves higher accuracy in comparison to the pure DL method, especially when using smaller fractions of the data, as shown in Fig. 3.

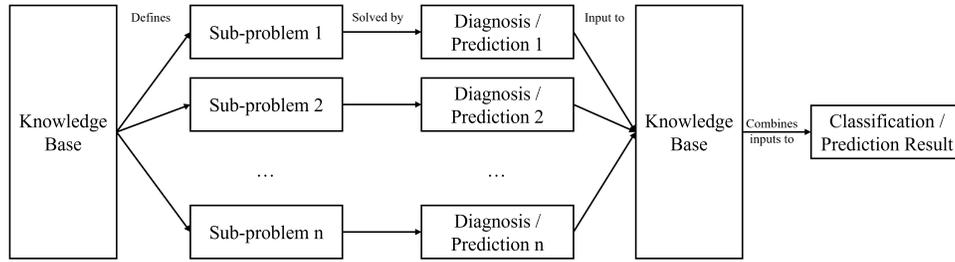


Figure 2. Process for splitting and combining DL tasks with knowledge.

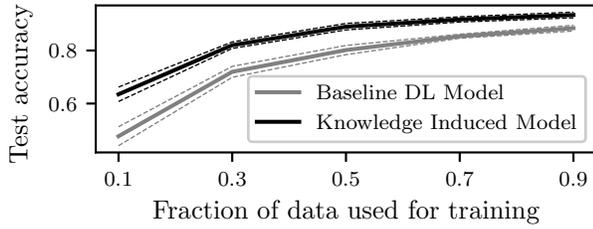


Figure 3. Results from the current (work-in-progress) paper, where we induce knowledge into DL models for bearing fault detection.

5. CONCLUSION

The proposed research direction for combining knowledge and DL for PHM has a lot of potential to improve current DL approaches in the PHM domain by taking the best of both worlds and alleviating the disadvantages of the individual approaches. The goal is to decrease the amount of labeled data needed, increase explainability and make models more robust towards outliers and minor domain changes. This year’s European Conference of the Prognostics and Health Management Society further underlines the importance and high relevance of these ideas for the PHM community by conducting a special session for inducing physics and domain expert knowledge in DL algorithms for PHM applications.

REFERENCES

Badreddine, S., Garcez, A. d., Serafini, L., & Spranger, M. (2022). Logic tensor networks. *Artificial Intelligence*, 303.

Cao, Q., Zanni-Merk, C., & Reich, C. (2018). Ontologies for manufacturing process modeling: A survey. In *International conference on sustainable design and manufacturing* (pp. 61–70).

Chao, M. A., Kulkarni, C., Goebel, K., & Fink, O. (2022). Fusing physics-based and deep learning models for prognostics. *Reliability Engineering & System Safety*, 217.

Cubillo, A., Perinpanayagam, S., & Esperon-Míguez, M. (2016). A review of physics-based models in prognos-

tics: Application to gears and bearings of rotating machinery. *Advances in Mechanical Engineering*, 8(8).

Fink, O., Wang, Q., Svensen, M., Dersin, P., Lee, W.-J., & Ducoffe, M. (2020). Potential, challenges and future directions for deep learning in prognostics and health management applications. *Engineering Applications of Artificial Intelligence*, 92.

Liao, L., & Köttig, F. (2014). Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. *IEEE Transactions on Reliability*, 63(1), 191–207.

Pawellek, G. (2016). *Integrierte Instandhaltung und ersatzteillistik: Vorgehensweisen, methoden, tools*. Springer-Verlag.

Selcuk, S. (2017). Predictive maintenance, its implementation and latest trends. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 231(9), 1670–1679.

Steenwinckel, B., De Paepe, D., Hautte, S. V., Heyvaert, P., Bentefrit, M., Moens, P., ... others (2021). Flags: A methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing expert knowledge with machine learning. *Future Generation Computer Systems*, 116, 30–48.

Wang, Q., Taal, C., & Fink, O. (2021, 07). Integrating expert knowledge with domain adaptation for unsupervised fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*.

BIOGRAPHY



Maximilian-Peter Radtke studied business mathematics at the University of Mannheim, Germany and graduated in 2018. After his studies he worked as a data science consultant in various industries for two and half years before returning to academia. Since 2021 he has been working at the Technische Hochschule Ingolstadt (THI) as part of AIMotion Bavaria and the research group AI applications for innovative production and logistic systems. His research interests include the combination of sym-

bolic and sub-symbolic AI approaches and the incorporation of knowledge into deep learning in the area of fault diagnostics and prognostics.



Jürgen Bock is a computer scientist, who graduated as Diplom-Informatiker from Ulm University, Germany, and as Bachelor of Information Technology with Honours from Griffith University, Brisbane, Australia, in 2006. He began his research career at the FZI Research Center for Infor-

mation Technology in Karlsruhe, Germany, and received his PhD from the Karlsruhe Institut of Technology (KIT) in 2012. After 2 years as post doc and team leader at the FZI, he joined the corporate research department of KUKA Robotics in Augsburg, Germany as developer and later leader of the team Smart Data and Infrastructure. In 2020 he joined the Technische Hochschule Ingolstadt (THI) as research professor in the area of AI applications in innovative production and logistics systems.

Index of Authors

- Abbasi, Ataollah , 432
Abreu, Fabio , 480
Al-Kahwati, Kammal , 1
Amin, Omnia , 9
Angelov, Plamen , 78
Antoni, Jérôme , 401
Aperstein, Yehudit , 146
Apostolidis, Asteris , 245
Asres, Mulugeta Weldezigina , 21
Béler, Cédric , 410, 590
Back, Thomas , 245
Baldo, Leonardo , 32
Baratchi, Mitra , 245
Barry, David , 580
Basci, Caner , 458
Berenji, Amirhossein , 43, 473
Berg, Mats , 269
Berri, Pier Carlo , 32
Bessa, Iury , 68
Bey-Temsamani, Abdellatif , 449
Birk, Wolfgang , 1
Bischof, Phillip , 49, 315
Blair, Jennifer , 58
Bock, Jürgen , 421, 594
Boekweit, Stan , 278
Borrelli, Elsi-Mari , 182
Bortman, Jacob , 376, 466
Boutrous, Khoury , 68
Brown, Blair , 9, 58
Camargos, Murilo , 78
Ceccarelli, Daniele , 182
Cerbah, Farid , 193
Cerisara, Christophe , 571
Chanthery, Elodie , 541
Chaoub, Alaaeddine , 571
Chin, Gareth Yen Ket , 480
Choi, Chihyeon , 286
Choi, Joo-Ho , 384
Cong, Jianli , 563
Cooper, Seth I. , 21
Cummings, Grace , 21
Dai, Junyan , 563
Dalla Vedova, Matteo D. L. , 32
Damsongsang, Prapanpong , 269
Danti, Piero , 87
de Pater, Ingeborg , 96, 278
del Moral, Pablo , 110
Deng, Weikun , 118, 574
Deodhar, Anirudh , 219
Dingeldein, Lorenz , 550
Dittmann, Jay , 21
Do, Phuc , 360, 586
Dubey, Abhishek , 480
Ezhilarasu, Cordelia Mattuvarkuzhali , 577
Fan, Ip-Shing , 509
Farago, François , 583
Federici, Fabio , 126
Fentaye, Amare , 136
Forbes, Alistair , 58
Francese, Arturo , 440
Fu, Yunxiao , 563
Gaffet, Alexandre , 541
Galeotta, Marco , 583
Garnier, Hugues , 193
Genc, Yakup , 211
Gildish, Eli , 146
Giossi, Rocco Libero , 269
Gogu, Christian , 118
Gore, Prayag , 556
Grebshtein, Michael , 146
Gryllias, Konstantinos , 338
Gupta, Ashit , 219
Hagmeyer, Simon , 156
Haslhofer, Bernhard , 175, 200
He, Yuning , 166
Heel, Robin , 200
Heistracher, Clemens , 175
Heitzinger, Clemens , 200
Helsen, Jan , 401
Hencken, Kai , 182
Hervé de Beaulieu, Martin , 193
Holly, Stephanie , 200
Holzner, Peter , 200
Huber, Lilach Goren , 530
Huber, Marco F. , 156
Hunemohr, David , 550
Ince, Kurçat , 211
Jung, Benoit , 360, 571, 586
Jacazio, Giovanni , 329
Jadhav, Vishal , 219
Jakobsson, Andreas , 401
Jalali, Anahid , 175
Jennions, Ian , 577
Jha, Mayank Shekhar , 193
Jia, Lilin , 577
Kamtsiuris, Alexander Athanasios , 231
Kandukuri, Surya T. , 368
Kandukuri, Surya Teja , 351
Kang, Jinlong , 239, 458
Katic, Denis , 200
Kaufmann, Thomas , 200
Kefalas, Marios , 245
Khukhunaishvili, Aleko , 21
Kim, Nam Ho , 384

- King, Stephen , 509, 580
 Kleemola, Jaakko , 261
 Klein, Renata , 376, 466
 Koçak, Gazi , 211
 Korkos, Panagiotis , 261
 Kovacs, Klaudia , 175
 Krivda, Andrej , 182
 Kulkarni, Rohan R. , 269
 Kundu, Pradeep , 556
 Kushnirski, Alex , 146
 Kyprianidis, Konstantinos , 136
 Laflamme, Catherine , 175
 Larsen, David , 126
 Le Cam, Mathieu , 126
 Lee, Jay , 556
 Lee, Juseong , 278
 Lee, Sangho , 286
 Lehtovaara, Arto , 261
 Linjama, Matti , 261
 Liu, Qi , 563
 Liwicki, Marcus , 306
 Lodes, Lukas , 294
 Lowenmark, Karl , 306
 Madar, Eyal , 466
 Maggiore, Paolo , 32
 Makienko, Igor , 146
 Mardt, Felix , 315
 Martin, Andrea De , 329
 Marx, Douw , 338
 Marzat, Julien , 583
 Mashhadi, Peyman Sheikholharam , 432
 Mathew, Manuel S. , 351
 Mayer, Julia , 449
 McArthur, Stephen , 9, 58
 Medjaher, Kamal , 118, 410, 574, 590
 Meisen, Tobias , 490
 Merle, Christophe , 541
 Metzler, Dirk , 49
 Mhangami, Washington , 580
 Minami, Takanobu , 556
 Minamino, Ryota , 87
 Mitici, Mihaela , 96, 278
 Morio, Jérôme , 118
 Mosallam, Ahmed , 239, 458
 Murata, Renato , 583
 Nejjari, Fatiha , 68
 Nguyen, Khanh T. P. , 118, 574
 Nguyen, Van-Thai , 360, 586
 Nilsen, Rune , 1
 Nilsfors, Evert Flygel , 1
 Nivre, Joakim , 306
 Noori, Nadia S. , 368
 Nowaczyk, Sławomir , 110, 432
 Ohana, Ravit , 376
 Omlin, Christian W. , 21, 351
 Ooijevaar, Ted , 449
 Palhares, Reinaldo M. , 68
 Parisi, Vincenzo , 329
 Park, Hyung Jun , 384
 Parygin, Pavel , 21
 Pashami, Sepideh , 110
 Pedersen, Jørgen F. , 392
 Peeters, Cédric , 401
 Piet-Lahanier, Hélène , 583
 Pizza, Gianmarco , 530
 Poupry, Sylvain , 410, 590
 Puig, Vicenç , 68
 Putz, Veronika , 449
 Qazizadeh, Alireza , 269
 Raddatz, Florian , 231
 Radtke, Maximilian-Peter , 421, 594
 Rahat, Mahmoud , 432
 Rawat, Anurag Singh , 480
 Ribot, Pauline , 541
 Roa, Nathalie Barbosa , 541
 Rognvaldsson, Thorsteinn , 432
 Ruiz-Carcel, Cristobal , 440
 Runkana, Venkataramana , 219
 Salaets, Rob , 449
 Sandin, Fredrik , 306
 Schall, Daniel , 200
 Schiendorfer, Alexander , 294
 Schlanbusch, Rune , 368, 392
 Schmidt, Immo , 550
 Schneeweiss, Jurgen , 175
 Schoeffl, Leopold , 200
 Schumann, Johann , 166
 Shanbhag, Vignesh V. , 368, 392
 Shen, Nannan , 239
 Simon, Henrik , 550
 Sobczak-Oramus, Karolina , 458
 Sol, Alon , 466
 Son, Youngdoo , 286
 Sorli, Massimo , 329
 Starr, Andrew , 440
 Stein, Bas van , 245
 Stephen, Bruce , 9, 58
 Stiftinger, Andreas , 200
 Sturm, Valentin , 449
 Suer, Alexander , 556
 Taal, Cees , 306
 Taco, John , 556
 Taghiyarrenani, Zahra , 43, 473
 Taheri, Atabak , 432
 Tang, Haichuan , 563
 Thielecke, Frank , 49, 315
 Thioulouse, Louis , 583
 Tian, Yin , 563
 Tonelli, Cecilia , 126
 Torchio, Marcello , 126
 Vadisala, Gautam Kumar , 480
 Varnier, Christophe , 239

Vichi, Giovanni , 87

Voisin, Alexandre , 360, 571, 586

Von Bulow, Friedrich , 490

Wang, Chengwei , 509

Wang, Yuan , 563

Weigert, Max , 521, 550

Wende, Gerko , 231

Youssef, Fares Ben , 239

Yu, David , 21

Yu, Huafeng , 166

Zaccaria, Valentina , 136

Zeiler, Peter , 156

Zerhouni, Noureddine , 239

Zraggen, Jannik , 530

Zhao, Xuejun , 563