Explainable Machine Learning Model for Predictive Maintenance in Smart Agricultural Facilities

M Kisten

orcid.org 0000-0001-7051-8575

Dissertation accepted in fulfilment of the requirements for the degree *Master of Science in Computer Science* at the North-West University

Supervisor: Prof AE Ezugwu

Co-Supervisor: Dr M Olusanya

Graduation: July 2024

Student number: 27701379

DECLARATION - AUTHORSHIP

I, Melvin Kisten, declare the following statements regarding my dissertation:

- Except where noted, this dissertation comprises solely my original research.
- No part of this dissertation has been submitted for any degree or examination at another university.
- I have not included data, images, charts, or information from other individuals in this dissertation without proper acknowledgement.
- This dissertation is free of text authored by others, except where duly cited. In instances where I have rephrased their ideas, I have provided references. Where I have used their exact words, I have enclosed them in quotation marks and cited them accordingly.
- This dissertation contains no text, illustrations, or tables directly extracted from the Internet, unless explicitly acknowledged and adequately cited in both the main text and the bibliography section.

Candidate: Melvin Kisten

Signature: _____

DECLARATION - PUBLICATIONS

This dissertation incorporates research that is critical to publications, either in the stages of preparation or published. Contributions of each author to these publications, encompassing experimental work and manuscript development, are listed below:

- Peer-reviewed journal article publication: Kisten, M., Ezugwu, A.E.S. & Olusanya, M.O. 2024. Explainable Artificial Intelligence Model for Predictive Maintenance in Smart Agricultural Facilities. *IEEE Access*, 12:24348-24367. 10.1109/ACCESS.2024.3365586

ACKNOWLEDGEMENTS

Immense gratitude is owed primarily to God Almighty for enabling this achievement.

Appreciation is extended to my supervisors, Prof. Absalom Ezugwu and Dr. Micheal Olusanya, who were astute mentors throughout. Their counsel stands as an embodiment of excellence. They have surpassed the prerequisites, ensuring the zenith of this research endeavour.

Gratitude is also extended to the North-West University and the CSIR for their unwavering support, research tutelage, and workshops. Their provision of diverse platforms for discourse, exhibitions, inquiries, and critiques has been pivotal in my development. The endeavour to acquaint themselves with me is duly recognised.

Family and friend networks also merit appreciation for their constant support and empathetic understanding. Heartfelt thanks are conveyed to all who played a role, large or small, in this research dissertation.

ABSTRACT

Machine Learning (ML) models in Smart Agricultural Facilities (SAF) often lack explainability, hindering farmers from taking full advantage of their capabilities. This study tackles this gap by introducing a model that combines a subset of eXplainable Artificial Intelligence (XAI), known as explainable Machine Learning, with Predictive Maintenance (PdM). The model aims to provide both predictive insights and explanations across four key dimensions: (1) data, (2) model, (3) outcome, and (4) end-user. This approach marks a shift in agricultural ML, reshaping how these technologies are understood and applied. The model outperforms related studies, showing quantifiable improvements. Specifically, the Long-Short-Term Memory (LSTM) classifier shows a 5.81% rise in accuracy. The eXtreme Gradient Boosting (XGBoost) classifier exhibits a 7.09% higher F1 score, 10.66% increased accuracy, and a 4.29% increase in Receiver Operating Characteristic-Area Under the Curve (ROC-AUC). These results could lead to more precise maintenance predictions in real-world settings. This study also provides insights into data purity, global and local explanations, and counterfactual scenarios for PdM in SAF. It advances ML by emphasising the importance of explainability beyond traditional accuracy metrics. The results confirm the superiority of the proposed model, marking a significant contribution to PdM in SAF. Moreover, this study promotes the understanding of ML in agriculture, emphasising explainability dimensions. Future research directions are advocated, including multi-modal data integration and implementing Human-in-the-Loop (HITL) systems aimed at improving the effectiveness of ML models and addressing ethical concerns such as Fairness, Accountability, and Transparency (FAT) in agricultural ML applications.

Keywords: Agriculture, Smart Agricultural Facilities, Predictive Maintenance, Machine Learning, Deep Learning, eXplainable Artificial Intelligence, explainable Machine Learning

LIST OF ABBREVIATIONS

Abbreviation	Definition
AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC	Area Under the Curve
CFE	Counterfactual Explanations
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CS	Computer Science
CSIR	Council for Scientific and Industrial Research
CSML	Cost-Sensitive Machine Learning
DiCE	Diverse Counterfactual Explanations
DL	Deep Learning
DNN	Deep Neural Network
DSR	Design Science Research
DT	Decision Tree
EL	Ensemble Learning
ELI5	Explain Like I Am Five
ELM	Extreme Learning Machine
FAT	Fairness Accountability Transparency
FN	False Negative
FP	False Positive
GB	Gradient Boosting

GBDT	Gradient Boosting Decision Tree
GBM	Gradient Boosting Model
GDPR	General Data Protection Regulation
GHz	Gigahertz
GPU	Graphics Processing Unit
HITL	Human-in-the-Loop
IAI	Interpretable Artificial Intelligence
ID	Identifier
IML	Interpretable Machine Learning
KDDC	Knowledge Discovery and Data Mining Cup
KNN	K-Nearest Neighbour
LGBM	Light Gradient Boosting Model
LIME	Local Interpretable Model-agnostic Explanations
LOCF	Last Observation Carried Forward
LR	Linear Regression
LRP	Layer-wise Relevance Propagation
LSTM	Long-Short-Term Memory
LVQ	Learning Vector Quantisation
MB	Megabyte
ML	Machine Learning
MLP	Multi-Layer Perceptron
NLP	Natural Language Processing
NN	Neural Network
OC	One-Class

OC-SVM	One-Class Support Vector Machine
OS	Operating System
PdM	Predictive Maintenance
PPS	Predictive Power Score
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
RAM	Random-Access Memory
ReLU	Rectified Linear Unit
RF	Random Forest
RFECV	Recursive Feature Elimination with Cross-Validation
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
ROC-AUC	Receiver Operating Characteristic-Area Under the Curve
RTX	Ray Tracing Texel eXtreme
RUL	Remaining Useful Life
SAF	Smart Agricultural Facilities
SHAP	SHapley Additive exPlanations
SLR	Systematic Literature Review
SRBD	Systematic Research on Big Data
STD	Standard Deviation
SVM	Support Vector Machine
TANH	Hyperbolic Tangent
TL	Transfer Learning
TM	Trademark
TN	True Negative

TP	True Positive
USA	United States of America
VT	Variance Thresholding
WoS	Web of Science
XAI	eXplainable Artificial Intelligence
XGB	eXtreme Gradient Boosting

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 Background of the Study

Agricultural enterprises, inherently driven by profit motives, face unique challenges (Bell *et al.*, 2014; Rana & Paul, 2017; Zhong *et al.*, 2021). Key among these are maintenance complexities and the high costs of farming machinery, which strain financial resources (Calcante *et al.*, 2013; Elahi *et al.*, 2022; Yildirim *et al.*, 2017; Zhou & Yin, 2019). These financial burdens hinder the acquisition and maintenance of essential equipment, affecting productivity. Farmers increasingly turn to Smart Agricultural Facilities (SAF) to counteract these pressures.

SAF transforms modern farming, transcending traditional methods. Innovations such as sensors, drones, robots, and advanced software (Eastwood *et al.*, 2019; Wolfert *et al.*, 2017) are not mere additions, but pivotal changes in agricultural practices. These state-of-the-art technologies streamline soil classification (Rahman *et al.*, 2018), intelligent pest management (Panchbhaiyye & Ogunfunmi, 2018), and efficient water leakage detection (Shankar *et al.*, 2020; Taravatrooy *et al.*, 2020); this technological evolution in agriculture does not just streamline processes; it fundamentally improves productivity and sustainability, marking a leap forward in farming efficiency.

Although the benefits of SAF are evident, these systems are not infallible and are prone to malfunctions. Addressing these limitations is crucial. The literature on Predictive Maintenance (PdM) provides valuable strategies to minimise downtimes and improve operational efficiency, thereby reducing farming costs (Vincent *et al.*, 2019); this approach addresses the immediate challenges and contributes to the long-term sustainability and cost-effectiveness of SAF in agriculture.

PdM in agriculture is not just about data collection; it is revolutionising equipment maintenance. By leveraging these data, farmers anticipate equipment failures, strategically improving operational efficiency, significantly reducing costs, and increasing productivity (Kande *et al.*, 2017). However, the success of PdM and SAF is dependent on critical factors: seamless data integration among stakeholders, acceptance of advanced technologies by users, and stringent adherence to privacy rights (Poppe *et al.*, 2015; Sonka, 2015). These elements are pivotal in determining whether such technological innovations will thrive or falter in the agricultural sector.

Driven by the challenges faced by PdM and SAF in agriculture, the need for explainable Machine Learning (ML) techniques is paramount, particularly in navigating challenges specific to these

fields (Doshi-Velez & Kim, 2017); this necessity is further underscored by legal mandates like the General Data Protection Regulation (GDPR), which require automated systems, such as those used in agriculture, to be explainable (Brauneck *et al.*, 2023; Chhetri *et al.*, 2022). Such regulations increase the importance of developing ML models that are not only effective but also comprehensible to users and compliant with legal frameworks (Selbst & Barocas, 2018).

This study confronts a fundamental question: How can one trust the predictions of an ML model without explanations, particularly for PdM in SAF? To address this, the study investigates the development of explainable ML models, with a specific emphasis on two key areas: strategies for PdM (Lughofer & Sayed-Mouchaweh, 2019) and methods for ML explainability (Doshi-Velez & Kim, 2017). The term explainable Machine Learning is deliberately chosen over eXplainable Artificial Intelligence (XAI) to reflect the study's concentrated focus on ML-specific algorithms and datasets, rather than the broader scope of Artificial Intelligence (AI) (Russell & Norvig, 2020); this distinction is crucial for clarity and precision in discussing the scope and results of the study.

1.2 Problem Statement

Recent ML models for PdM in SAF often fail to elucidate their decision-making processes, a critical flaw (Orn *et al.*, 2020); this lack of explainability is compounded by a prevalent focus in the literature on reporting only accuracy and F1 scores, neglecting to provide comprehensive train and test results, thereby compromising both transparency and robustness. Furthermore, the ambiguous role of ML in agriculture, whether as a support or a substitute for farmers, remains unresolved (Harmani *et al.*, 2022). For these reasons, farmers struggle to understand the current ML models used in agriculture and cannot take full advantage of their capabilities. Addressing these gaps requires the application of explainable ML to improve the understanding and usability of PdM in SAF (Ehsan *et al.*, 2021; Hepenstal *et al.*, 2021; Riedl, 2019; Sperrle *et al.*, 2021). Therefore, this study proposes a model that merges explainable ML with PdM in the context of SAF to improve transparency and practical applicability.

1.3 Aim and Objectives of the Study

Embarking on this study involves conducting a thorough Systematic Literature Review (SLR) bolstered by a bibliometric analysis. The goal is to grasp the current landscape of ML models, focusing on explainable ML. Furthermore, the task encompasses reviewing existing literature on both Deep Learning (DL) and Machine Learning (ML) implementations for PdM, with a dedicated

emphasis on explainable ML in this context. These literary explorations provide detailed insights into the crucial need for explainable ML, especially concerning the opaque decision-making processes in ML models in SAF (Orn *et al.*, 2020).

Moreover, the literature surveys serve as catalysts for shaping the study's objectives and inform the development and evaluation of the combined model; they underscore the significance of explainable ML in agricultural Predictive Maintenance. Consequently, the study offers a comprehensive overview of explainable ML.

Following a rigorous evaluation, which involves implementing and evaluating four DL algorithms and seven ML algorithms. Furthermore, due to the prevalent usage of the F1 score and for comparative analysis, it is used to select the most outstanding DL and ML algorithms in each dataset, which are then used for the explainability stage. The overarching aim is to (1) predict maintenance needs and (2) provide explanations using explainable ML for PdM in SAF.

To achieve the research aim, the specific research objectives are summarised as follows:

i. Perform a Systematic Literature Review employing bibliometric analysis to assess the contemporary status of ML models and ascertain the demand for explainable Machine Learning.

ii. Develop ML models, particularly focusing on Machine Learning and Deep Learning, to predict maintenance requirements.

iii. Evaluate the effectiveness of the formulated Machine Learning and Deep Learning models.

iv. Uncover and elucidate the rationale behind the predictions generated by the proposed ML model.

1.4 Research Questions

In line with the set objectives, this research seeks to address the following four research questions:

i. What is the current landscape of ML models and the need for explainable Machine Learning?

ii. How should one develop an ML model to predict maintenance needs?

iii. How does the proposed ML model perform?

iv. What and why does the proposed ML model make specific predictions?

1.5 Research Motivations

Although some studies have previously considered ML explainability, they predominantly adopt a generic, one-size-fits-all approach to explainability (Molnar *et al.*, 2020). The focal point of this research effort is to cultivate a more nuanced and adaptable approach to explainability tailored to diverse stakeholder needs. Unlike generic explanations often used, this study aims to align the granularity of explanations with the distinctive requirements of different stakeholders, acknowledging the distinct needs of an ML specialist versus those of a farmer. The rationale for this approach is rooted in the recognition that explanation purposes and utilities vary, necessitating a more tailored and versatile framework. In particular, this study accentuates the value of explanations that cater to scientific insight and the internal requirements of stakeholders, contributing to a more nuanced and informed application of ML in agricultural PdM.

1.6 Research Contributions

This study contributes substantially to the ML field by conducting a thorough SLR and a bibliometric analysis review, providing a deep exploration of the existing landscape of ML models. In particular, it underscores the critical need for explainable ML; it moves beyond the conventional focus of classification accuracy that characterises many related studies. By adopting a more transparent approach, this study reports both training and testing results; it includes the robust Receiver Operating Characteristic-Area Under the Curve (ROC-AUC) metric, offering a stricter assessment of the DL and ML algorithms used.

In addressing the opaqueness of ML models used for PdM in SAF, where explanations of their decision-making techniques remain absent (Orn *et al.*, 2020), this study underscores the integration of explainable ML and PdM in SAF; these contribute to four explainability dimensions: (1) data, (2) model, (3) outcome, and (4) end-user (Arrieta *et al.*, 2020; Doshi-Velez & Kim, 2017). Moreover, these four dimensions of explainability indicate a much-needed paradigm shift toward explanations that should align with the expectations of diverse stakeholders. In particular, this study is the pioneer in exploiting this combined model in the agricultural sector, comprehensively addressing the four dimensions of explainability.

The theoretical contribution of this study lies in its adaptation of design knowledge from various domains, such as explainable ML and PdM, specifically tailored to suit the SAF context. Furthermore, including a comprehensive SLR and literature review not only strengthens the theoretical foundations of the research but also lays a solid groundwork for future research in this evolving field. The specific contributions of this study are summarised as follows:

- Pioneers the integration of explainable ML and PdM in SAF from four explainability dimensions: (1) data, (2) model, (3) outcome, and (4) end-user.
- Provides an SLR and literature review on the current landscape of ML models and the need for explainable ML.
- Reports transparent training and testing results with the stricter ROC-AUC metric, emphasising the urgency of explainable ML beyond accuracy.
- Adapts design knowledge from explainable ML and PdM to suit the SAF context.
- Strengthens the theoretical foundations and future directions of explainable ML for PdM in SAF.

1.7 Scope and Limitations of the Study

The scope of the study was to incorporate explainable ML and PdM in SAF, focusing on (1) machines and (2) irrigation systems. The study aimed to predict potential maintenance requirements before the actual failure of such systems while offering explanations for these predictions using explainable ML. However, the study faced limitations due to the lack of PdM data and only using two time series datasets. Moreover, PdM results were limited, focusing solely on DL and Ensemble Learning (EL) for ML models, steered by relevant literature. Similarly, explainable ML results were confined to (1) data, (2) model, (3) outcome, and (4) end-user explanations, guided by dimensions of explainability extracted from notable literature (Arrieta *et al.*, 2020; Doshi-Velez & Kim, 2017). Finally, the bibliometric analysis was also constrained; details of these restrictions reside in the “Review methodology” section of the bibliometric analysis. For replication and ensuring reproducibility, the researcher's experimental code is available in their GitHub repository at: https://github.com/iammelvink.

1.8 Layout of the dissertation

This dissertation is structured as follows:

Chapter 2 engages with the literature by employing a bibliometric analysis as an integral SLR component; this process transcends a mere perusal of related literature; it constitutes a purposeful endeavour to accentuate the imperative for explainable ML. The aim is not solely to elucidate but to focus on the role of explainable ML models in PdM. The distinctive aspect lies in its ability to pinpoint ML models tailored for PdM. It is not a mere synopsis, but a synthesis of results. It offers a comprehensive view of the current topography and the potential trajectory of explainable ML models in agricultural PdM.

Chapter 3 illuminates the integration of DL and ML algorithms in achieving the first facet of the research aim: predicting maintenance needs. All DL and ML algorithms employed in this study trace their origins to the reviewed literature; this chapter meticulously expounds on DL and ML, while the forthcoming chapter concentrates on the application of explainable ML for PdM. Thus, this chapter expands on using DL and ML in PdM, specifically grounded in the knowledge from the reviewed scholarly literature.

Chapter 4 deals with the operational facets of explainable ML techniques, a pivotal step in actualising the second facet of the research aim: providing explanations using explainable ML for PdM in SAF. The focus is directed explicitly towards PdM, unfolding the details inherent in deploying explainable ML. The chapter aims to discuss the four dimensions of explainability: (1) data, (2) model, (3) outcome, and (4) end-user (Arrieta *et al.*, 2020; Doshi-Velez & Kim, 2017). The elucidation of these dimensions transcends mere theoretical discourse; it expands on the frameworks used to achieve explainability for DL and ML. Post-hoc explainable methodologies assume prominence in model, outcome, and end-user dimensions, deployed after the model undergoes training (Arrieta *et al.*, 2020). On the contrary, the data dimension adopts a composite methodology, encompassing both post-hoc and pre-hoc (before model training) approaches. Tools such as Deepchecks, SHAP, and DiCE assume centrality as the instrumentalities of the study, facilitating the realisation of the four explainability dimensions. Concurrently, the proposed model takes centre stage, poised to demonstrate its efficacy.

In Chapter 5, the stage is set to reveal the research methodology. Prepare for the experimental design's intricacies, predicting maintenance needs, and providing explanations using explainable ML for PdM

in SAF. No aspect is overlooked, as the chapter meticulously discloses the study's foundational elements: the research paradigm, a robust theoretical framework, the designated datasets, ethical considerations, a pivotal research design, and the meticulous system and parameter configuration to achieve the research aim.

Chapter 6 elucidates the results and discussion, interlacing discernments from meticulous data analysis. The emphasis on PdM and explainable ML sharpens, disentangling these aspects into precise sub-themes. Prognostics and diagnostics assume prominence, predicting system and component malfunctions, facilitated by the adept deployment of DL and ML algorithms. A study on "data explainability" adopts a data-centric perspective, analysing the intricacies and anticipations embedded in data insights. Global explainability advances, unveiling the layers of the model's conduct through model-centric lenses in the "model dimension". Concurrently, local explainability takes the forefront, illuminating singular prediction instances and enriching the "outcome dimension". Navigating the "end-user dimension", the focus is on formulating explanations that attain a balance between abstraction and detail. The discussion traverses a comparative analysis with related studies, including thematic terrains (prognostics, diagnostics, data, model, outcome, and end-user), elucidating implications and intertwining results with the broader literature framework.

Chapter 7 concludes the dissertation and similarly weaves the empirical results with the formulated research objectives and questions, aligning with the overarching research aim. It emphasises pivotal contributions and delineates a trajectory for prospective research endeavours and recommendations. The work established in previous chapters culminates in this section, addressing the identified research gaps and advocating for a paradigmatic transformation in the implementation of ML in view of SAF.

CHAPTER 2 LITERATURE REVIEW

2.1 Overview

This study carried out a comprehensive literature review. The aim was to critically analyse the latest ML-driven PdM techniques (Figure 2.1). The study sought to identify explainable ML models designed for PdM, focusing on their role in improving maintenance practices. Moreover, the review aimed to provide valuable insights into the strengths, limitations, and applicability of different explainable ML models for PdM. By evaluating and synthesising the results, it endeavoured to offer a broad overview of the current landscape and potential route of explainable ML models in the context of agricultural PdM.

2.2 Predictive Maintenance

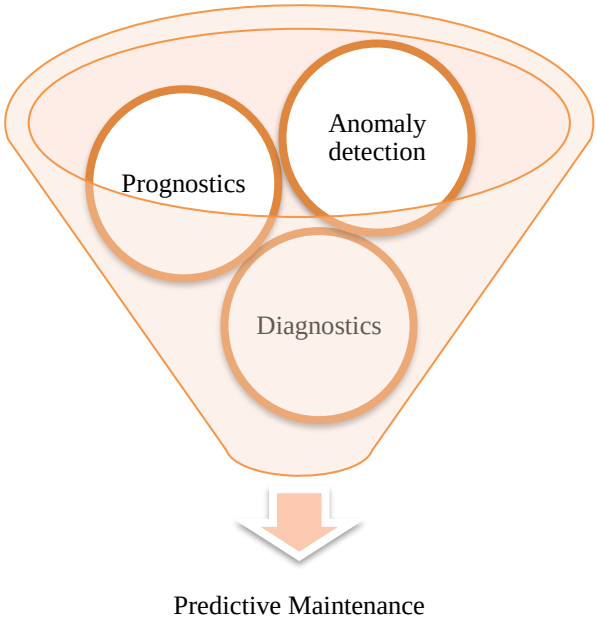

Figure 2.1: Approaches to Predictive Maintenance

2.2.1 Approaches to Predictive Maintenance

In this review, three approaches to PdM are revealed: (1) anomaly detection, (2) prognostics, and (3) diagnostics (Lughofer & Sayed-Mouchaweh, 2019). Anomaly detection is like finding a needle in a haystack, where the needle represents unusual or unexpected patterns in the data. Prognostics

predict the future performance or condition of a system or component. Diagnostics involves identifying problems or issues in a system or process by analysing its current performance and behaviour. Of the reviewed literature, 11 studies focused on prognostics (Garouani *et al.*, 2022; Gashi *et al.*, 2023; Ghasemkhani *et al.*, 2023; Hajgató *et al.*, 2022; Hermansa *et al.*, 2021; Jakubowski *et al.*, 2021; Kononov *et al.*, 2023; Kuzlu *et al.*, 2020; Serradilla *et al.*, 2021; Upasane *et al.*, 2021; Wu *et al.*, 2021), three on anomaly detection (Choi *et al.*, 2022; Langone *et al.*, 2020; Mey & Neufeld, 2022), and two on prognostics and diagnostics (Serradilla *et al.*, 2021; Steurtewagen & Van den Poel, 2021).

Consequently, the reviewed literature did not concentrate on integrating two elements: anomaly detection and diagnostics approaches. Moreover, none of the reviewed studies exclusively focused on diagnostics approaches. The absence of studies that thoroughly consider diagnostics alone is not just a gap; it is a chasm that needs research that does not just look at these methods in passing. Instead, it analyses them with the seriousness they deserve. Studies must focus on this integration, exploring how anomaly detection and prognostics can seamlessly lead to effective diagnostics, thus enhancing the robustness and efficiency of PdM in SAF.

2.3 Deep and Machine Learning for Predictive Maintenance

For prognostics, the use of Recurrent Neural Networks (RNNs) and Long-Short-Term Memory (LSTM) techniques proved to be a recurrent theme (Kononov *et al.*, 2023; Wu *et al.*, 2021); this was substantiated by a study that engaged RNNs and LSTMs to categorise machine conditions at specific time points (Wu *et al.*, 2021). The dataset used had 20480 observations and 35 attributes captured at 10-minute intervals. Their model demonstrated an accuracy of 90.07%.

Similarly, in predicting the Remaining Useful Life (RUL) in prognostics research, the use of Bidirectional Recurrent Neural Networks (BiRNNs) and LSTMs were observed (Kononov *et al.*, 2023). The distinguishing factor lies in the bidirectional nature of these models, enabling them to consider the trajectory of information in both the positive and negative directions. The dataset included 20631 observations and 26 attributes. Moreover, the study incorporated a sliding window, a methodology to incorporate intervals in a dataset. Opting for precision over accuracy, the study evaluated the model's capacity to predict a specific class accurately. They achieved a precision of 96.15%. In the context of another study that used LSTMs, they applied them to an anomaly detection problem (Choi *et al.*, 2022). A distinctive aspect of this study was the implementation of LSTMs in an unsupervised use case.

Another frequently used ML algorithm in prognostics was the One-Class Support Vector Machine (OC-SVM) (Choi *et al.*, 2022; Hermansa *et al.*, 2021; Serradilla *et al.*, 2021); this was highlighted by a research effort that leveraged the OC-SVM for outlier detection, aiming to reduce false alarms in the context of PdM (Hermansa *et al.*, 2021). Their study used two synthetic datasets of 30000 observations for each of them. Their results revealed a substantial reduction in false alarms, 90.25% on average.

When tackling prognostics and anomaly detection, two other studies used the OC-SVM (Choi *et al.*, 2022; Serradilla *et al.*, 2021). Both research efforts trained the ML algorithm through sensor data obtained from machines in the manufacturing industry. A striking divergence in approach was observed; One study presented the problem as unsupervised anomaly detection (Choi *et al.*, 2022), while the other study (Serradilla *et al.*, 2021) framed it as supervised prognostics.

In addition, the SVM showed utility in another instance, where it was used to solve a prognostics problem through AutoML (Garouani *et al.*, 2022). The underlying premise of AutoML is its ability to simplify the application of ML for researchers by reducing the need for extensive coding, thus democratising access to ML techniques for non-expert users (Reif *et al.*, 2014). A noteworthy distinction in this research lies in the SVM's application for multi-class classification, deviating from previously reviewed literature (Garouani *et al.*, 2022).

Despite the prevalent use of the OC-SVM in PdM, as highlighted in reviewed studies (Choi *et al.*, 2022; Hermansa *et al.*, 2021; Serradilla *et al.*, 2021), the One-Class Support Vector Machine (OC-SVM) possesses notable limitations in the ML research context. The inherent design of the OC-SVM restricts its capabilities to unsupervised ML problems, making it unsuitable for the scope of the current study, which is a supervised ML problem.

In addition, the Random Forest (RF) algorithm also emerged as a popular tool for prognostics, as evidenced in numerous studies (Garouani *et al.*, 2022; Gashi *et al.*, 2023; Kuzlu *et al.*, 2020); this was shown in the case of a study that applied the RF to understand how components interact with each other and the degree of impact each has on the corresponding components regarding failure (Gashi *et al.*, 2023). The dataset included 8761 observations and 6 attributes. This study stood out because it focused on prognostics at the level of individual components, diverging from the traditional approach of addressing the entire machine or system. Moreover, the study boasts an improved F1 score, indicating that the model was better at exactness and completeness on

imbalanced data. On average, the model boasts a 7% increase compared to the model that tackles prognostics at the system level.

An alternative study that used RF in prognostics also incorporated AutoML, demonstrating the ease of access to ML for non-expert users (Garouani *et al.*, 2022). However, it must be acknowledged that AutoML's lack of customisability can pose a challenge in optimising model performance, marking it as a noteworthy limitation of AutoML (Reif *et al.*, 2014).

The flexibility of the RF was shown in an instance where it functioned as a regressor instead of a classifier, thus forecasting continuous values (Kuzlu *et al.*, 2020); this research applied the RF to a prognostics issue in the smart grid sector, aiming to forecast energy usage with a dataset of 13 attributes. The distinguishing factor was that this was the only reviewed study that tackled the smart grid sector using prognostics.

Like RF, Ensemble Learning (EL) has found utility in the prognostics area of manufacturing industries (Garouani *et al.*, 2022; Steurtewagen & Van den Poel, 2021). The EL models resemble the RF models, employing multiple prediction models. However, the EL differentiates itself by incorporating different types of models, in contrast to RF, which uses multiple models of the same kind, such as numerous Decision Trees (DT); this is shown by a research effort that used the Gradient Boosting (GB) ensemble model as part of an AutoML implementation (Garouani *et al.*, 2022). In another study, the researchers used eXtreme Gradient Boosting (XGBoost) (Steurtewagen & Van den Poel, 2021). The study also used the F1 score as a performance metric, just as in a previously reviewed study (Gashi *et al.*, 2023). Furthermore, the study achieved an F1 score of 93.0%, an accuracy of 90.0%, and a ROC-AUC of 91.10% (Steurtewagen & Van den Poel, 2021).

Based on Bayes's theorem, Balanced K-Star, an uncommon ML algorithm in prognostics, is similar to the K-Nearest Neighbour (KNN) algorithm (Ghasemkhani *et al.*, 2023). It stands out because it was explicitly designed to deal with imbalanced data. Similarly underrated in prognostics is the Multi-Layer Perceptron (MLP), a variant of Neural Networks (NNs) (Upasane *et al.*, 2021). Furthermore, in the sphere of NNs, Learning Vector Quantisation (LVQ), an example of an Artificial Neural Network (ANN), emerged as also being uncommon (Brinkrolf & Hammer, 2018).

Other attempts to approach prognostics from less common angles were Extreme Learning Machine (ELM) and Transfer Learning (TL) (Serradilla *et al.*, 2021). The research treated the issue as a semi-supervised problem, showing some unlabelled data points. The premise of ELM, a variation of feedforward Neural Networks (NNs), merits attention due to its enhanced training speed compared to traditional NNs (Huang *et al.*, 2006). Furthermore, a study incorporated TL, a technique that facilitates the transfer of acquired knowledge from one model to another, thus allowing further training on new data (Shin *et al.*, 2016). TL enabled the model to be further trained on a new dataset.

Deep Convolutional AutoEncoders, which are born from NNs, also held a position in the domain of prognostics, as emphasised in recent studies (Hajgató *et al.*, 2022; Jakubowski *et al.*, 2021). A particular study used a dataset comprising 32 attributes from the manufacturing sector to address prognostic challenges (Jakubowski *et al.*, 2021). Another study followed a similar path, focusing on detecting vulnerable machine components in the same sector (Hajgató *et al.*, 2022). A striking finding across these studies was the widespread application of DL algorithms. However, these algorithms are being criticised because of their high complexity and lack of transparency.

In the field of PdM, DL algorithms were found to be commonly used for anomaly detection (Choi *et al.*, 2022; Mey & Neufeld, 2022); this was shown in the case of a study that used a dataset with 31 attributes (Choi *et al.*, 2022). Moreover, the study was applied to the manufacturing sector, which, from previously reviewed studies, is a popular sector for PdM.

The issue of diagnostics in PdM presented a notable finding. Two studies ventured into diagnostics together with prognostics (Serradilla *et al.*, 2021; Steurtewagen & Van den Poel, 2021). Moreover, the study used outlier detection to pinpoint issues in a system (Serradilla *et al.*, 2021). However, another study used sensor data to achieve diagnostics (Steurtewagen & Van den Poel, 2021). A notable finding was that none of the reviewed studies focused exclusively on diagnostics. Therefore, after a thorough review of PdM, the following section leads to an explainable ML for PdM.

2.4 Explainable Machine Learning

In the fast-paced era of DL and ML, sophisticated model deployment now pervades sectors like healthcare, finance, and agriculture. However, the intricate nature of these models often clouds their decision-making processes, raising concerns about transparency (Orn et al., 2020). This opacity has catalysed the emergence of explainability in DL and ML, a concept that transcends mere transparency.

Explainability entails dissecting the complexities of the DL and ML models to render their decision-making understandable, catering to both experts and non-experts. Crucially, explainability embodies a spectrum of facets, each pivotal to demystifying and ensuring the reliability of the DL and ML models. Figure 2.2 shows these aspects: (1) data, (2) model, (3) outcome, and (4) end-user, highlighting their integral roles in fostering an understandable, reliable DL and ML explainability framework.

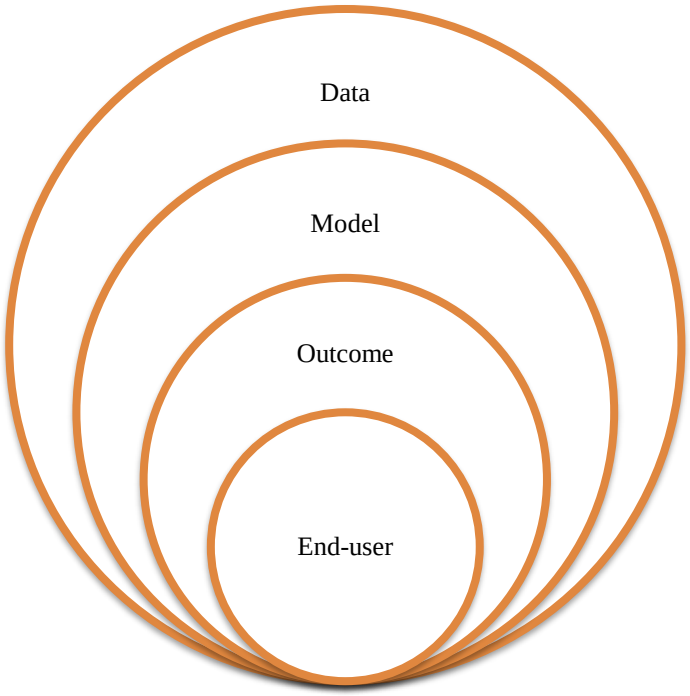

Figure 2.2: Dimensions of explainability

2.4.1 Dimensions of explainability

In this review, four key dimensions are highlighted: (1) data, (2) model, (3) outcome, and (4) end-user. Here, the focus zooms in on the "data dimension." This dimension's role is pivotal; it is about peering into data's dark corners to identify its inherent shortcomings and the researcher's pipe dreams regarding its potential. It is not just about skimming the surface to see what is possible with the data, but drilling down to understand its true capabilities and limits in delivering meaningful explainability (Doshi-Velez & Kim, 2017). However, the reviewed studies woefully neglect to dig into these limitations. They sidestep critical questions: Can the data sustain the weight of the insights one is trying to extract from it? This is not just a gap; it is a gaping hole in research that calls for more rigorous exploration.

Studies that acknowledge and wrestle with these limitations are needed to provide a clearer picture of what the data can and cannot tell in the area of PdM in SAF.

Second, the "model dimension" seeks to unveil how input data sway the predictions formed by the model (Doshi-Velez & Kim, 2017). An essential presumption that a researcher should hold is that features or attributes maintain independence and bear no connection to one another, signifying that one feature remains unaffected by the others; this presumption, however, does not always hold. Hence, the "model dimension" remains susceptible to bias. Despite this, all but three studies incorporated this explainability dimension (Gashi *et al.*, 2023; Ghasemkhani *et al.*, 2023; Kononov *et al.*, 2023). Moreover, two studies used the "model and outcome dimensions", which the text considers next (Garouani *et al.*, 2022; Kuzlu *et al.*, 2020).

Outcome explainability is crucial in enabling individuals to understand the rationale behind specific predictions made by an ML model (Doshi-Velez & Kim, 2017). A thorough literature review found that only two studies (Jakubowski *et al.*, 2021; Upasane *et al.*, 2021) were exclusively dedicated to exploring outcome explainability; this scarcity of research efforts dedicated to the "outcome dimension" highlights a need to amplify research endeavours in this area. By expanding research in the outcome dimension, one can address the existing gap and develop a deeper understanding of how to provide meaningful explanations for the predictions produced by ML models. Such advances would significantly enhance ML models' overall explainability and interpretability, fostering transparency, trust, and informed decision-making.

Within the fourth dimension, which pertains to the "end-user", the primary importance is creating explanations that strike the right balance between abstraction and detail to cater to the target audience (Arrieta *et al.*, 2020); this dimension aims to foster trustworthy and credible ML systems, specifically focusing on explainability for non-technical users. Notably, none of the studies reviewed addressed this dimension, specifically in unveiling explanations for non-technical individuals.

2.4.2 Approaches to explainability

Six explainability approaches emerge from the literature (Guidotti *et al.*, 2018). In pursuit of (1) local explainability, one seeks to clarify a single prediction instance. On the other hand, (2) global explainability concentrates on unveiling the model's behaviour. Within the reviewed literature, two studies followed both local and global explainability (Mey & Neufeld, 2022; Wu *et al.*, 2021), while the remaining 13 endeavoured to deliver local explainability alone (Choi *et al.*, 2022;

Garouani *et al.*, 2022; Gashi *et al.*, 2023; Ghasemkhani *et al.*, 2023; Hajgató *et al.*, 2022; Hermansa *et al.*, 2021; Jakubowski *et al.*, 2021; Kononov *et al.*, 2023; Kuzlu *et al.*, 2020; Langone *et al.*, 2020; Serradilla *et al.*, 2021; Steurtewagen & Van den Poel, 2021; Upasane *et al.*, 2021). None of the reviewed literature exclusively targeted global explainability. Thus, a prevailing need for further research in global explainability arises.

Furthermore, other approaches employed encompass (3) model-specific strategies confined to particular ML models and (4) model-agnostic approaches applicable universally to any ML model. The review discovered that ten studies employed model-agnostic methods (Choi *et al.*, 2022; Garouani *et al.*, 2022; Gashi *et al.*, 2023; Ghasemkhani *et al.*, 2023; Hermansa *et al.*, 2021; Jakubowski *et al.*, 2021; Kononov *et al.*, 2023; Langone *et al.*, 2020; Serradilla *et al.*, 2021; Steurtewagen & Van den Poel, 2021), while three used model-specific techniques (Hajgató *et al.*, 2022; Upasane *et al.*, 2021; Wu *et al.*, 2021) and only two studies harnessed both (Kuzlu *et al.*, 2020; Mey & Neufeld, 2022). These findings underscore the pressing need for additional research exploring integrating these two approaches (model-agnostic and model-specific).

Two additional strategies are the (5) model-centric and (6) data-centric approaches. First, model-centric methods aim to dissect the interplay between input characteristics and target outcomes in the model. At the same time, the data-centric approach probes into the data used to train the model (Arrieta *et al.*, 2020); this perspective determines the data's consistency, proper curation, and suitability for addressing the core issue. Furthermore, data and concept drift, data profiling, and adversarial resilience form the framework of data-centric explainability strategies. The reviewed literature showed that all reviewed studies were dedicated to model-centric approaches, and none focused on data-centric approaches. For this reason, a noteworthy gap arises, requiring further studies that emphasise data-centric explainability approaches.

2.5 Explainable Machine Learning for Predictive Maintenance

Model explainability often employs the SHapley Additive exPlanations (SHAP) method (Choi *et al.*, 2022; Garouani *et al.*, 2022; Gashi *et al.*, 2023; Hermansa *et al.*, 2021; Kononov *et al.*, 2023; Kuzlu *et al.*, 2020; Serradilla *et al.*, 2021; Steurtewagen & Van den Poel, 2021); this was shown by a study that explained predictions by illustrating the degree of impact each attribute or feature exerts on the ultimate prediction (Hermansa *et al.*, 2021). Similarly, SHAP found application in the model dimension in the manufacturing sector (Serradilla *et al.*, 2021). The study leveraged explainability for diagnostic objectives, providing explanations for predictions as a diagnostic tool.

An alternative study also used SHAP on sensor data in the manufacturing sector (Steurtewagen & Van den Poel, 2021). The dataset included compressor units, plant monitoring, and inspection reports. The reviewed literature revealed SHAP as a common method to tackle model explainability. However, as observed by a comparative study, SHAP provides comprehensive information and results, which may be intimidating for non-expert users (Kuzlu *et al.*, 2020). Thus, it becomes essential to present SHAP's results in a manner that is intuitive for the end-user.

In addition to SHAP, Local Interpretable Model-agnostic Explanations (LIME) emerged as another common method for model explainability (Kuzlu *et al.*, 2020; Mey & Neufeld, 2022); this was shown in the case of a study that applied LIME to models that were used for anomaly detection in a PdM use case (Mey & Neufeld, 2022). The study explained the DL algorithms used in transportation datasets. Moreover, despite LIME's ability to be used across any ML algorithm, it excels in local explanations, targeting specific predictions without encompassing the model's global behaviour.

Another technique used in model explainability is Layer-wise Relevance Propagation (LRP) (Mey & Neufeld, 2022; Wu *et al.*, 2021); this was evident in the research that applied LRP to models designed for prognostics on a time series dataset (Wu *et al.*, 2021). LRP was used in DL algorithms as it best suited them. Moreover, the study showed how both time step-wise and feature-wise inputs affect the prediction, reinforcing the effectiveness of LRP on DL algorithms.

When addressing the explainability of the model and outcome dimensions, a comparative study applied LIME, SHAP, and Explain Like I Am Five (ELI5) (Kuzlu *et al.*, 2020). The study showed feature attribution for each predictor. Moreover, it presented feature attribution for a range of predictions. Furthermore, the study highlighted the time efficiency of LIME, SHAP, and ELI5, exposing LIME as the fastest model to train. At the same time, ELI5 provided intuitive explanations (Kuzlu *et al.*, 2020). However, one drawback of ELI5 is that it is not model-agnostic, which limits it to specific ML algorithms.

Turning attention to the quest for outcome explainability, the Counterfactual Explanations (CFE) framework is a common method (Garouani *et al.*, 2022; Jakubowski *et al.*, 2021); this was shown in the case of a study that applied CFEs to increase the acceptability of ML (Garouani *et al.*, 2022). It is worth highlighting that the study used AutoML, which makes ML easily accessible to non-expert users but lacks customisability (Reif *et al.*, 2014); this shortcoming introduces a difficulty in optimising model performance. Therefore, after a thorough review of explainable ML for PdM,

the subsequent section provides a bibliometric analysis to present the current landscape of ML models and the need for explainable ML.

2.6 Bibliometric analysis

This study conducted a detailed bibliometric analysis that required transparency in ML. The analysis presents the methodology used, followed by data-related processes, descriptive statistics, keyword analysis, knowledge synthesis, conceptual structure, and social structure.

2.6.1 Review methodology

Searches were done in Titles, Abstracts, and Subject Headings. Also, the search was tailored to encompass literature from 2012 to 2022, excluding 2023, as it is an incomplete year at the time of writing. Moreover, note that in the subsequent databases, "Keywords Plus" and "Keywords" function as Subject Headings. The benefit of searching through Subject Headings lies in analysing synonyms, domain-specific terms, and pertinent keywords, thereby boosting the search to find relevant literature concerning the research topic (Garfield, 1990). The themes extracted from the literature review also motivated the final search strings. Table 2.1 shows the exact search strings used in both databases: Web of Science (WoS) and Scopus.

Table 2.1: Databases and search strings

Database	Search string
Web of Science (WoS)	(TS=(explain* OR interpret* OR xai OR iai) AND TS=(ai OR ml) AND TS=(maintenance OR detect* OR prognos* OR diagnos*)) AND (PY==("2012" OR "2013" OR "2014" OR "2015" OR "2016" OR "2017" OR "2018" OR "2019" OR "2020" OR "2021" OR "2022") AND DT==("ARTICLE") AND LA==("ENGLISH") AND SJ==("COMPUTER SCIENCE"))
Scopus	(TITLE-ABS-KEY (explain* OR interpret* OR xai OR iai) AND TITLE-ABS-KEY (ai OR ml) AND TITLE-ABS-KEY (maintenance OR detect* OR prognos* OR diagnos*)) AND PUBYEAR > 2011 AND PUBYEAR < 2023 AND (LIMIT-TO (SUBJAREA , "COMP")) AND (LIMIT-TO (DOCTYPE , "ar")) AND (LIMIT-TO (LANGUAGE , "English")) AND (LIMIT-TO (SRCTYPE , "j"))

Also, note the parameters selected in the search strings in Table 2.2.

Table 2.2: Search parameters

Parameter	Value
Timespan	2012-2022
Language	English
Subject area	Computer Science
Document types	Article
Sources	Journals

Figure 2.3 depicts a Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) flow diagram (Page *et al.*, 2021), outlining the data collection, screening, and analysis steps and how resources were chosen or excluded for the review. The "Identification" stage involved gathering 1711 records from two academic databases (WoS and Scopus), with 361 duplicates removed. Screening assessed each record's relevance, leading to the exclusion of 51 preprints and non-journal records, based on criteria in Table 2.2. The "Included" stage refers to the final selection of 759 records for review.

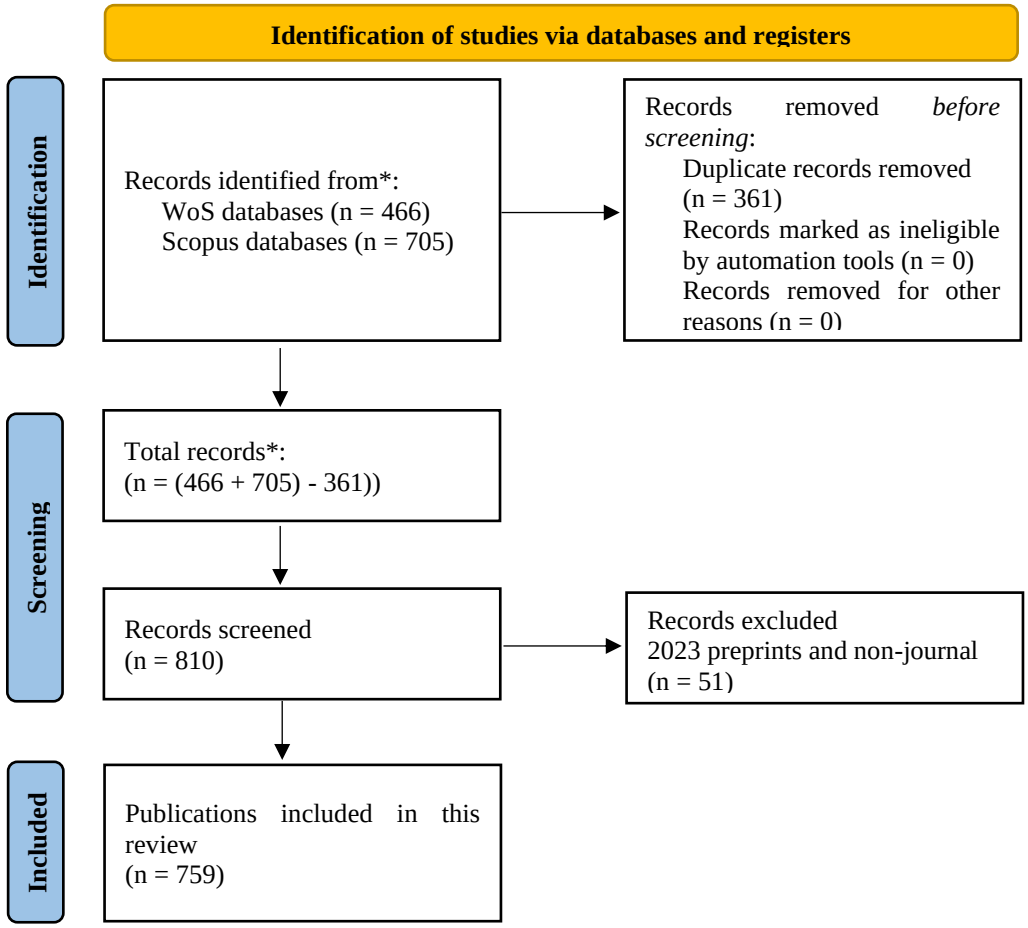

Figure 2.3: Summary of the data extraction and screening process

2.6.2 Data collection

In conducting a rigorous bibliometric analysis, the study extracted information from two academic databases: WoS and Scopus. The extraction of the said data transpired on the 30th of June, 2023, maintaining uniformity through the utilisation of an identical search string on both databases; this systematic approach included 759 publications in this review, providing substantial material for this bibliometric analysis.

2.6.3 Data integration

The study required merging data from the WoS and Scopus databases; this integration was executed using the Bibiometrix R software package (Aria & Cuccurullo, 2017) to merge the two

datasets. In addition, the "mergeDbSources" function was used for the merge. Committed to data integrity, the "remove.duplicated" feature ensured the elimination of any remaining duplication.

2.6.4 Data pre-processing

The data were pre-processed to eliminate errors and inconsistencies (Luengo *et al.*, 2020). Filtering of specific terms such as "na", "n/a", "n.a", and "0" reinforced the accuracy of the data. Moreover, the pre-processing extended to managing synonyms, a technique that enhanced the consistency of the data. The term in each row in the column “Used” replaced the subsequent terms in the column “Replaced” in the section “Additional Tables and Figures” of the Appendices.

2.6.5 Data analysis

The study secured comprehensive bibliometric data from numerous publications in conducting a bibliometric and critical analysis. Using RStudio, an advanced research tool developed by Posit, the study enhanced and streamlined the analysis process (Posit, 2023). Using Bibliometrix, a specialised package in R software, enabled the researcher to perform the analysis (Aria & Cuccurullo, 2017).

2.6.6 Descriptive statistics

In the bibliometric analysis, 759 documents were included, comprised solely of journal articles. The average document age was 2.16 years, each accruing an average citation count of 14.24. Furthermore, the yearly expansion rate was 44.85%, accounting for 35274 references. The document contents included 3878 instances of "Keywords Plus" (Garfield, 1990) and 2564 instances of "Author Keywords". An overview of the final dataset used for the bibliometric analysis is also provided in Table 2.3.

Table 2.3: Overview of the final dataset (2012 – 2022)

Description	Results
Timespan	2012:2022
Sources (Journals, Books, among others.)	298
Documents	759
Annual growth rate%	44.85
Average age of the document	2.16

Average citations per doc	14.24
References	35274
Document contents	
Keywords Plus	3878
Author's Keywords	2564
Authors	
Authors	3559
Authors of single-authored docs	24
Authors' Collaboration	
Single-authored docs	24
Co-Authors per Doc	5.43
International co-authorships%	20.42
Document types	
Article	759

2.6.7 Keyword analysis

Keywords act as a crucial connector, linking researchers to relevant resources in a database. These terms reveal the central notions of a research topic, offering a common language for its explanation. Without appropriate keywords, searching for pertinent documents becomes a formidable task. An assembly of such terms, recognised as the most illustrative of their work by the authors themselves, constitutes the author's keywords. As illustrated in Figure 2.4, the dataset for this study unveiled the top 10 author keywords featured in the documents. Notably, the keyword "machine learning" appeared as the most commonly used term with 212 instances, in line with the context of explainable ML in PdM research. The subsequent keywords were "explainable artificial intelligence" and "deep learning", with frequencies of 208 and 145, respectively. Keywords provided by the author serve as a powerful tool, efficiently dissecting the knowledge structure inherent in scientific disciplines.

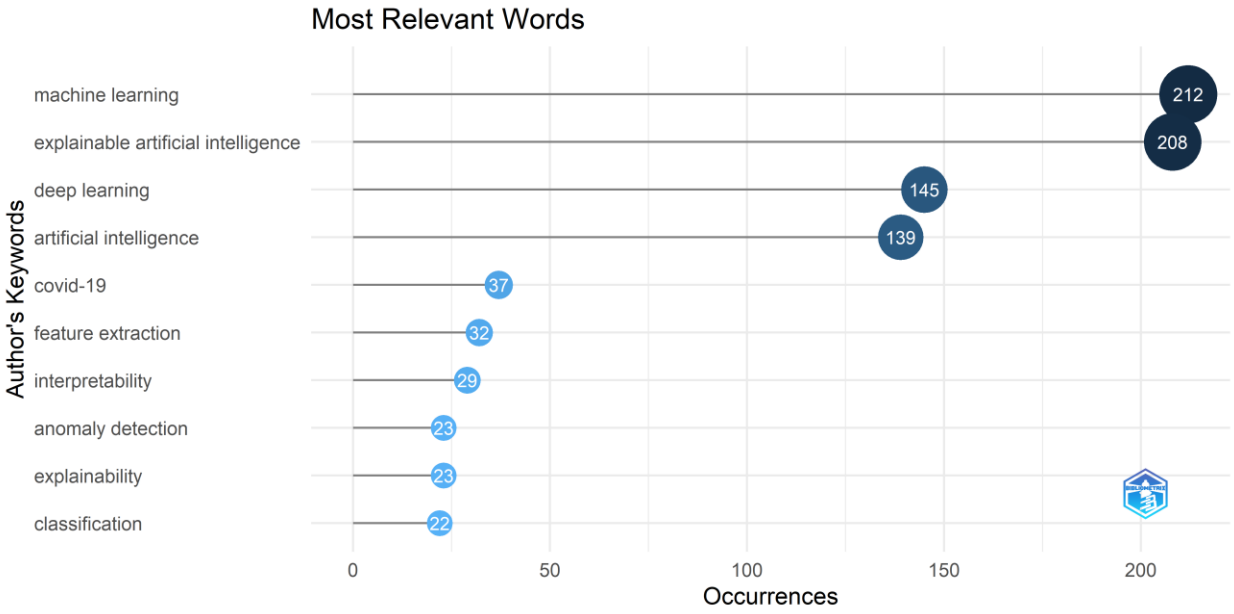

Figure 2.4: Top 10 most frequent author keywords in the dataset

2.6.8 Knowledge synthesis and Conceptual structure

Science mapping is an intellectual pursuit to reveal the interconnected network within the continually changing landscape of scientific knowledge (Cobo *et al.*, 2011). Fundamentally, it works to disclose both the structural elements and the dynamic aspects of scientific research, viewed through the prism of knowledge synthesis. Therefore, science mapping equips its practitioners with a statistical viewpoint, allowing them to probe the depths of the scientific knowledge domain.

Unveiling the crucial themes and developments, the conceptual structure served as a strategic objective. The interactive co-occurrence grid applied the Walktrap algorithm (Pons & Latapy, 2006), leveraging association as its standardisation protocol, as depicted in Figure 2.5. Words, when merged into a unified document, become interlinked. The use of this logical framework casts light on the topics covered within a research domain and the specific insights these topics encapsulate. Moreover, it underscores the evolutionary trajectory pursued by studies over time. Figure 2.5's co-occurrence network traces the developmental path of explainable ML in PdM research. The network of connections illustrates a picture of interconnected knowledge.

Consequently, the terms "explainable artificial intelligence" and "deep learning" frequently appeared in the literature; this shows the necessity for DL algorithms to incorporate elements of explainability.

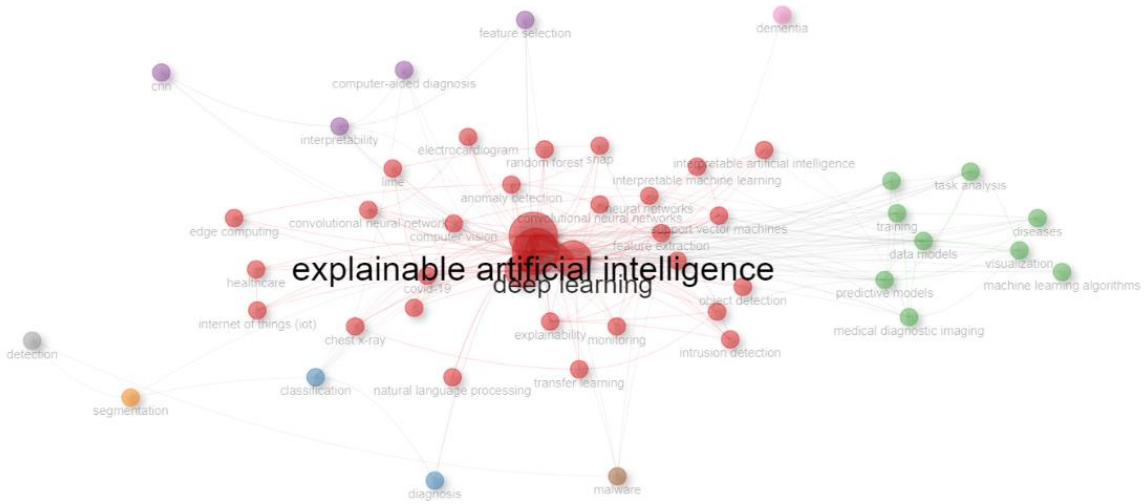

Figure 2.5: Co-occurrence network

2.6.9 Social structure

The findings illustrated in Figure 2.6 showed Wang Y and Zhang Y as key contributors, possessing substantial collaboration scores. However, it is pertinent to note that their collaborative efforts did not extend to one another.

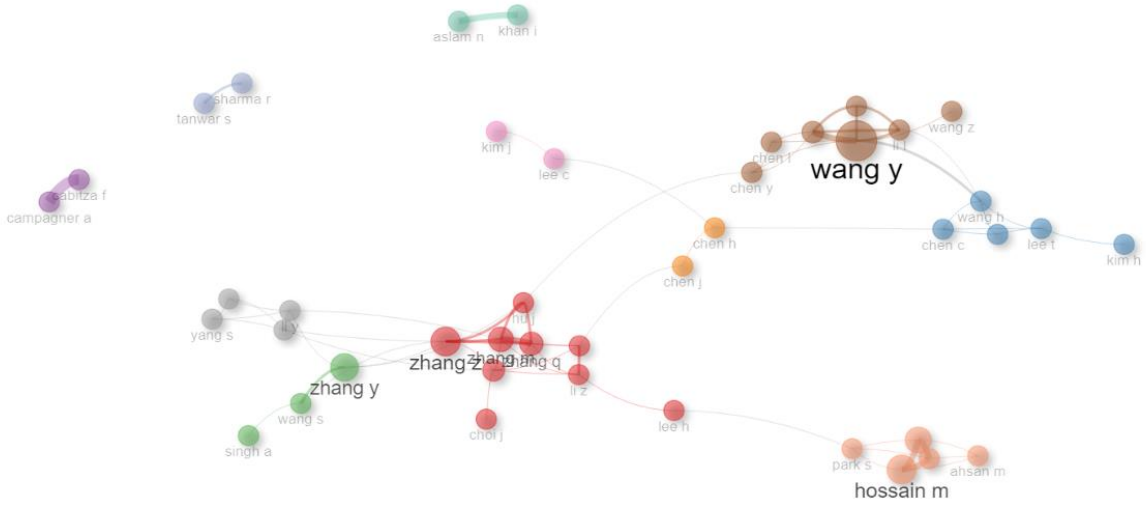

Figure 2.6: Collaboration network by author

Figure 2.7 further clarified that respected institutions such as the University of Florida, Tsinghua University, and Central South University maintain notable collaboration ranks, although their collaborative efforts did not intersect.

Figure 2.7: Collaboration network by institution

Similarly, Figure 2.8 shows that the United States of America (USA) collaborated predominantly with China and India. Therefore, after a thorough bibliometric analysis, the following section will present the research gap and critical analysis to highlight the reasons for the current study.

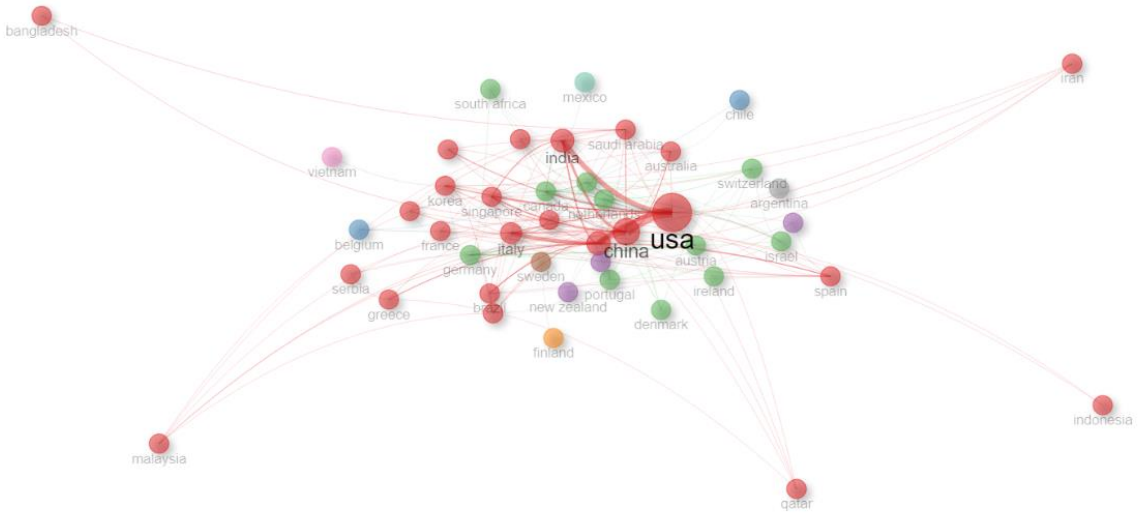

Figure 2.8: Collaboration network by country

2.7 Research gap

The analysis of research gaps carries significance as it unveils what is lacking in the literature, illuminating the primary rationales propelling this study, and highlighting contributions. Furthermore, the rigorous critique showcases the sectors where explainable ML is needed. Subsequently, this critique underscores the prevalent explainability techniques, an influential factor in steering this study's choice of methodologies. Similarly, bibliometric analysis helped to direct the requirement for explainable ML. The following are details of the critical analysis, and the summary provides a conclusion for the chapter.

2.7.1 Critical analysis

The review unveiled three sectors where PdM was used with explainable ML: (a) transportation, (b) manufacturing, and (c) smart grids. Among these, manufacturing garnered the attention of twelve studies (Choi *et al.*, 2022; Garouani *et al.*, 2022; Gashi *et al.*, 2023; Ghasemkhani *et al.*, 2023; Hajgató *et al.*, 2022; Hermansa *et al.*, 2021; Jakubowski *et al.*, 2021; Langone *et al.*, 2020;

Serradilla *et al.*, 2021; Steurtewagen & Van den Poel, 2021; Upasane *et al.*, 2021; Wu *et al.*, 2021), while transportation claimed two (Kononov *et al.*, 2023; Mey & Neufeld, 2022), and smart grids only one (Kuzlu *et al.*, 2020).

The literature employed various state-of-the-art models for PdM. EL models emerged as prevalent, particularly with tabular data (Garouani *et al.*, 2022; Steurtewagen & Van den Poel, 2021). Similarly, DL models encompassed RNNs, LSTMs, and NNs (Kononov *et al.*, 2023; Upasane *et al.*, 2021; Wu *et al.*, 2021). Consequently, this study shall harness similar model types, to achieve comparable results.

Concerning the metrics used to evaluate the proposed model, the PdM models in the reviewed literature used the accuracy metric (Wu *et al.*, 2021). Likewise, the literature also used the precision metric (Kononov *et al.*, 2023). Furthermore, another metric used was the F1 score, which is suitable for imbalanced data, representing the harmonic mean of precision and recall metrics (Gashi *et al.*, 2023; Steurtewagen & Van den Poel, 2021). Despite the lack of its appearance within almost all of the reviewed literature, the ROC-AUC also bears value for evaluating models on data with imbalanced classes, showing the model's ability to classify each class (Carrington *et al.*, 2023; Steurtewagen & Van den Poel, 2021; Wang & Yao, 2013). Therefore, this study shall employ the F1 score, accuracy, and ROC-AUC. Furthermore, due to the prevalent usage of the F1 score and for comparative analysis, it will be used to select the most outstanding DL and ML algorithms on each dataset, which will then be used for the explainability stage.

In addition, the literature review emphasised popular explainability techniques such as ELI5, LIME, SHAP, and LRP for the model and outcome dimensions. Unfortunately, ELI5 is confined to tree-based and parametric-linear models (Kuzlu *et al.*, 2020). Specifically, the objective of ELI5 is to provide weights for individual features, which conveys the extent of impact each has on the final predictions. Moreover, while LIME is faster than ELI5 and SHAP, it falls behind in presenting distributions. Furthermore, SHAP delivers full global explanations (Choi *et al.*, 2022; Garouani *et al.*, 2022; Gashi *et al.*, 2023; Hermansa *et al.*, 2021; Kononov *et al.*, 2023; Kuzlu *et al.*, 2020; Serradilla *et al.*, 2021; Steurtewagen & Van den Poel, 2021). Likewise, SHAP provides detailed visualisations in explanation plots. Meanwhile, LRP is best suited for DL models (Mey & Neufeld, 2022; Wu *et al.*, 2021).

Furthermore, although Counterfactual Explanations (CFE) show their application in the outcome dimension (Garouani *et al.*, 2022; Jakubowski *et al.*, 2021), this study will embrace them to address

the end-user dimension; this is done by providing a user with options or counterfactuals to know what to change to get the desired outcome. Similarly, another matter of concern is the data dimension. Regrettably, reviewed studies did not address this facet by unearthing attainable degrees of explainability from the data (Doshi-Velez & Kim, 2017). Consequently, this study will amplify "data dimension" efforts.

The bibliometric analysis revealed a prevailing trend of standard ML models being employed in PdM, substantiating the literature review finding (Choi *et al.*, 2022; Garouani *et al.*, 2022; Gashi *et al.*, 2023; Ghasemkhani *et al.*, 2023; Hermansa *et al.*, 2021; Kuzlu *et al.*, 2020; Serradilla *et al.*, 2021; Steurtewagen & Van den Poel, 2021) (refer to Figure 2.4). Moreover, it underscored the prevalent use of explainable ML in DL, reinforcing the imperative need for explainability in DL models (refer to Figure 2.5). The social structure also showed an increasing demand for collaborative efforts to advance explainable ML in PdM research (refer to Figures 2.6, 2.7, and 2.8).

Several notable gaps are observed across numerous dimensions in the domain of explainable ML and PdM. These dimensions include the facets of data explainability, the outcome aspect, and the end-user-geared explainability. Likewise, there is an overarching mindset toward global explainability and the combination of model-specific and model-agnostic methods. Delving further into PdM, there is a distinct vacuum concerning integrating anomaly detection and diagnostics methods. In addition, the absence of the diagnostic approach exclusively. Furthermore, the literature review points to a lack of coverage of the intersection of explainable ML and PdM in the context of SAF; this further exacerbates why farmers struggle to understand the current ML models used in agriculture and cannot take full advantage of their capabilities.

Therefore, this study proposes a model that merges explainable ML and PdM to (1) predict maintenance needs and (2) provide explanations for the predictions made by the model. Trained on a time series dataset of maintenance records, sensor readings, and (a) machine and (b) water pump statuses, the proposed ML model will predict potential failures (prognostics), highlighting their root causes by component level (diagnostics), and then address explainability from four dimensions: (1) data, (2) model, (3) outcome, and (4) end-user (Arrieta *et al.*, 2020; Doshi-Velez & Kim, 2017).

2.8 Summary

This chapter delivered a thorough literature review on cutting-edge ML models and the demand for explainable ML. It aimed to analyse the most recent ML-fuelled PdM methodologies in the agricultural sphere. In addition, it pinpointed explainable ML models crafted for PdM, spotlighting their contribution to enhancing maintenance strategies and operational efficiency. The review provided invaluable perspectives on the strengths, constraints, and suitability of various explainable ML models for PdM. By assessing and synthesising the results, it presented a comprehensive panorama of the existing terrain and the prospective trajectory of explainable ML models in agricultural PdM. The next chapter presents the application of DL and ML for PdM in achieving the research aim.

CHAPTER 3 APPLICATION OF DEEP AND MACHINE LEARNING FOR PREDICTIVE MAINTENANCE

3.1 Overview

Recall that this study carried a dual aim: first, predicting maintenance needs, and second, supplying explanations using explainable ML for PdM in SAF; this segment probes into expanding on the application of DL along with ML for tackling the initial section of the aim. The subsequent chapter shall tackle the latter part. Consequently, this chapter seeks to broaden the understanding of DL and ML in PdM and their implementation in this study. It should also be noted that all DL and ML algorithms employed in this study drew inspiration from reviewed literature.

3.2 Deep Learning

3.2.1 Artificial Neural Network

Computational models known as ANNs draw inspiration from the structure and functionality of human brains. Organised into layers, these models comprise interlinked processing units, called neurons. Foundational work, dating back to the 1940s, paved the way for the principles of ANNs. In this context, the essential structure of ANNs was born through the proposal of a simple neuron model (McCulloch & Pitts, 1943). The conception included binary threshold units, receiving inputs, and yielding binary outcomes based on established thresholds, which shape current ANN designs.

The perceptron model presented in 1958 marked another noteworthy contribution. Acting as an artificial neuron, this perceptron model holds utility for binary classification tasks (Rosenblatt, 1958); it laid the groundwork for more intricate NN architectures. Other frequently referenced materials from 2012 gave rise to Deep Neural Networks (DNNs), which tackle more intricate issues and became the standard architecture for modern NNs and DNNs currently used (Hinton *et al.*, 2012).

3.2.2 Convolutional Neural Network

The Convolutional Neural Network (CNN) is integral to DL. Its composition encompasses numerous layers, notably convolutional, pooling, and fully connected. CNNs are designed to enable autonomous hierarchical representation learning from input data. The concept of CNNs highlighted their ability to learn features directly from the data (Lecun *et al.*, 1998).

Another ground-breaking study accentuated deep CNNs' ability in image classification tasks, marking an unprecedented advancement (Krizhevsky *et al.*, 2017). Further contributing to understanding, another paper unveiled how CNNs learn and encapsulate features, providing critical insights into their intricate operation (Zeiler & Fergus, 2013).

3.2.3 Bidirectional Recurrent Neural Network and Long-Short-Term Memory Neural Network

Incorporating a Bidirectional Recurrent Neural Network (BiRNN) entails using a Neural Network architecture that processes input data in both forward and reverse directions. The BiRNN design empowers the model to integrate forthcoming contextual data during its training, ultimately sharpening its aptitude for understanding sequences proficiently (Schuster & Paliwal, 1997); this proves especially invaluable in tasks where understanding the entire sequence's context is paramount.

Long Short-Term Memory (LSTM) is a Recurrent Neural Network form. It emerged as a solution to the vanishing gradient predicament encountered in standard RNNs, facilitating the modelling of extensive dependencies in sequential data. LSTMs attain this feat by implementing a sophisticated computational unit (Hochreiter & Schmidhuber, 1997); this unit integrates mechanisms for selective remembrance and forgetfulness of information across arbitrary time spans.

3.2.4 Convolutional Long-Short-Term Memory Neural Network

A Convolutional LSTM melds a CNN and LSTM, tailored for handling tasks intertwining spatial and temporal data (Shi *et al.*, 2015). The integral elements comprising Convolutional LSTM encompass convolutional layers responsible for distilling spatial attributes from data inputs such as image or video frames. They use filters to analyse the data, identifying patterns. Second, LSTM layers are employed to model sequential data while capturing time-dependent connections. These layers possess memory cells that store and retrieve data over elongated sequences. Merging the CNN and LSTM, convolutional layers typically precede LSTM layers; this strategy empowers the network to distill spatial attributes before modelling temporal dependencies in the identified features.

3.3 Machine Learning

3.3.1 Decision Tree

Employing the Decision Tree (DT) classifier facilitates classification tasks (Breiman *et al.*, 1984). Deriving a DT model enables data classification into distinct classes. DTs represent supervised ML algorithms, as do all classifiers used in this study. Moreover, the DT is used because it is what the RF is built upon.

3.3.2 Random Forest

Part of the ensemble family, the Random Forest employs numerous DT classifiers to generate predictions (Breiman, 2001). The meta-estimator, Random Forest, fits DT classifiers onto diverse sub-samples in the dataset. Through averaging, it improves predictive power and mitigates overfitting.

3.3.3 Bagging

This method finds its origins in diverse literature sources. When data subsets are randomly selected from samples, they are referred to as Pasting (Breiman, 1999). If samples are drawn with replacement, it goes by Bagging (Breiman, 1996). When random subsets pertain to features rather than samples, they adopt the title Random Subspaces (Ho, 1998). Furthermore, when base estimators are constructed using subsets comprising samples and features, they earn the moniker Random Patches (Louppe & Geurts, 2012). Thus, the Bagging classifier enhances predictive accuracy by training base classifiers on randomly selected data subsets. Then, it combines their predictions to boost performance.

3.3.4 Adaptive Boosting

AdaBoost, an abbreviation for Adaptive Boosting, is an EL technique that unites numerous feeble learners into one formidable learner (Freund & Schapire, 1997; Hastie *et al.*, 2009). The process commences by training a classifier on the initial dataset. Then, it replicates the classifier and retrains it on identical data, although with alterations in the weights of inaccurately classified instances. These adjustments steer subsequent classifiers toward more intricate cases.

3.3.5 eXtreme Gradient Boosting

Implementing gradient boosting machines with efficiency and scalability manifests in XGBoost, a term synonymous with eXtreme Gradient Boosting. Moreover, XGBoost is recognised for its rapid and robust performance and its success became evident during KDDCup 2015 (Chen & Guestrin, 2016). It was the tool chosen by each winning team among the top 10.

3.3.6 Light Gradient Boosting Model

Primarily used for ranking and classification tasks, the Light Gradient Boosting Model (LightGBM) propels the high-performance distributed gradient boosting framework (Ke *et al.*, 2017); this framework, rooted in DT algorithms, excels in efficiency, making it potent for numerous ML applications. LightGBM notably accelerates traditional Gradient Boosting Decision Trees (GBDT) training regimen by more than 20-fold, maintaining near-identical accuracy.

3.3.7 Categorical Boosting

CatBoost, a gradient-boosting ensemble technique, harnesses the power of multiple DTs for accurate predictions. Through iterative refinements, this algorithm effectively mitigates the loss function (Dorogush *et al.*, 2018). The term "Cat" signifies “categorical”, underlining CatBoost's proficiency with categorical features. Although it shares its place with algorithms such as XGBoost and LightGBM under the gradient boosting umbrella, it distinguishes itself through its natural compatibility with categorical features (XGBoost has also added this compatibility starting from version 2), eliminating the requirement for exhaustive pre-processing. Its prominence emerges when managing tabular data.

3.4 Prognostics and Diagnostics

Taking advantage of the discussed DL and ML algorithms, this study presents and discusses two critical aspects of PdM: prognostics and diagnostics. Successful prediction of system failure prognostics is achieved, mirroring this success in diagnostics, where component failure is predicted.

3.5 Preamble for explainability

In PdM, avoiding explainability pitfalls is crucial. For instance, explainability hinges on a model's ability to generalise effectively (Molnar *et al.*, 2020). Thus, the quality of explainability directly

reflects the performance of the underlying model. Also, if the model experiences underfitting or overfitting, it jeopardises the explanation's integrity. So, what measures does this study take to improve the efficacy of explainability methods? This study selects only the most robust DL and ML algorithms for explainability. Rigorous pre-processing is conducted, involving multiple steps rather than one. To address imbalanced classes, Cost-Sensitive Machine Learning (CSML) is employed instead of techniques that alter the original data distribution (Khan *et al.*, 2018; López *et al.*, 2013); this technique mitigates class imbalances by assigning weights relative to the number of instances in each class. In addition, overfitting is proactively countered.

3.6 Summary

This study set forth a two-part aim: first, (1) predicting maintenance needs and, subsequently, (2) supplying explanations using explainable ML for PdM in SAF. The chapter has endeavoured to discuss DL and ML towards achieving part one of the aim. In contrast, the succeeding chapter focuses on attaining the second part of the aim. Thus, this section served to elaborate on the use of DL and ML in PdM, explicitly rooted in insights gleaned from the reviewed literature. The next chapter presents the application of explainable ML for PdM.

CHAPTER 4 APPLICATION OF EXPLAINABLE MACHINE LEARNING FOR PREDICTIVE MAINTENANCE

4.1 Overview

This chapter intends to expound on using explainable ML in PdM, in response to the second part of the research aim: supplying explanations using explainable ML for PdM in SAF. Also, to detail the techniques employed to attain the four dimensions of explainability: (1) data, (2) model, (3) outcome, and (4) end-user (Arrieta *et al.*, 2020; Doshi-Velez & Kim, 2017). Moreover, the proposed model is highlighted.

Furthermore, consider that model, outcome, and end-user explainability dimensions are achieved through post-hoc explainability approaches; this means the that explanation method is applied following model training (Arrieta *et al.*, 2020). On the other hand, the data dimension opts for a combined approach, incorporating both post-hoc and pre-hoc (before training a model).

4.2 Data explainability

4.2.1 Analysing data purity

When training a model, emphasis typically goes into data pre-processing, involving actions like identifying duplicates or unique values, eliminating noise and unwanted data, spotting outliers, managing missing values, addressing mixed data types, and performing feature engineering to enhance ML models (Jiang *et al.*, 2018). These methods aim to eliminate impurities from the training data. However, what happens if a black-box ML model is trained on data with reduced purity and yields subpar performance? This requires data purity analysis using the Predictive Power Score (PPS), a crucial step in addressing the data explainability dimension; this evaluation reveals the potential of a single feature to accurately predict the target variable.

i. Deepchecks Python framework

Deepchecks encompass extensive verifications analysing data challenges (Chorev *et al.*, 2022). These verifications, designed for various models and data types such as tabular, Natural Language Processing (NLP), and vision, offer straightforward customisation and expansion. Each verification seeks to identify specific potential pitfalls. Moreover, Deepchecks boasts numerous pre-established verifications for analysing data purity.

Consequently, this study used Deepchecks to perform data explainability and gauge the expectations of the data used. Furthermore, the explanations provided are notably technical and tailored to expert stakeholders.

4.3 Model and Outcome explainability

SHAP, conceived in 2017 (Lundberg & Lee, 2017), marked an evolution from Shapley values, originating from game theory in 1952 (Shapley, 1952); this tool aims to attribute feature effect towards an ML model's final prediction. Consider a basketball team looking to understand why they emerged victorious in a game. In explainable ML, each player's performance parallels a feature in the model, with the game outcome being the prediction. SHAP resembles a postgame analysis, determining each player's (feature's) contribution to the victory (prediction). It assigns a "contribution score" to each player (feature) based on their actions during the game, aiding in understanding their impact on the final result. Similarly, SHAP assigns contribution values (SHAP values) to features in an ML model, explaining why a specific prediction was made. It assists in understanding the effect of each feature on the model's decision, akin to the basketball team analysing players' roles in their win.

Table 4.1 shows a comparative analysis of SHAP and feature importance. SHAP values distribute each feature's impact on a prediction, providing a detailed insight into how each feature affects outcomes. They employ cooperative game theory to allocate values based on marginal attributions, which can be positive or negative, indicating the feature's direction of effect (Shapley, 1952). SHAP values offer global and local feature significance, evaluating all data points and targeting specific predictions (Lundberg & Lee, 2017). They adhere to the Shapley axioms, guaranteeing fairness and feature attribution consistency. However, they can be computationally intensive for extensive datasets, but find robust support in various ML algorithms.

Conversely, feature importance rates each feature based on its contribution to the model's performance, ranking features by importance but lacking a nuanced understanding of individual feature impact (Molnar, 2022). Various methods, such as Gini, Permutation importance, or Gain, calculate importance scores, typically non-negative, signifying the impact direction but not the magnitude. Feature importance primarily focuses on global feature significance and may not always align with axiomatic principles, potentially resulting in inconsistencies. Nevertheless, it is faster and computationally less demanding, widely implemented in most ML algorithms, and primarily used for feature selection.

Table 4.1: Comparative analysis of SHAP and Feature importance

Aspect	SHAP	Feature importance
Definition	Ensures fair share	Rates model impact
Interpretation	Unveils effects	Ranks by significance
Methodology	Applies game theory	Employs calculations
Values	Signals impact	Shows direction
Global vs Local	Offers both scales	Primarily global
Consistency	Honours axioms	May lack consistency
Implementation	Resource-intensive	Faster and efficient
Algorithm support	Widely backed	Commonly adopted
Use cases	Deciphers complex models	Guides feature selection

Consequently, SHAP is used to perform model and outcome explainability, enabling the extraction of explanations from global and local perspectives tailored to meet the needs of expert stakeholders.

4.4 End-user explainability

Developed to generate varied counterfactual explanations, Diverse Counterfactual Explanations (DiCE) use determinantal point processes (Mothilal *et al.*, 2020). Its model-agnostic disposition adds to its appeal. Interpreting counterfactual explanations does not need complex assumptions, with clarity at the forefront. Altering feature values according to counterfactuals leads to changes in prediction, aligning them with predefined predictions (Wachter *et al.*, 2018). Far from any mysterious happenings behind the scenes, it presents multiple counterfactual explanations per instance, offering an array of options for the end-user to select.

As seen in Figure 4.1, counterfactuals reveal a relationship similar to a "what if" scenario, linking input X (feature) denoted X1, X2, X3, and Xn to the “outcome” (prediction). Counterfactual explanations of a prediction identify the minimal alteration required in feature values, shifting the prediction toward an established outcome (Molnar, 2022). Moreover, counterfactuals offer explanations appealing to human understanding, as they contrast the existing instance and exhibit selectiveness in concentrating primarily on limited feature modifications.

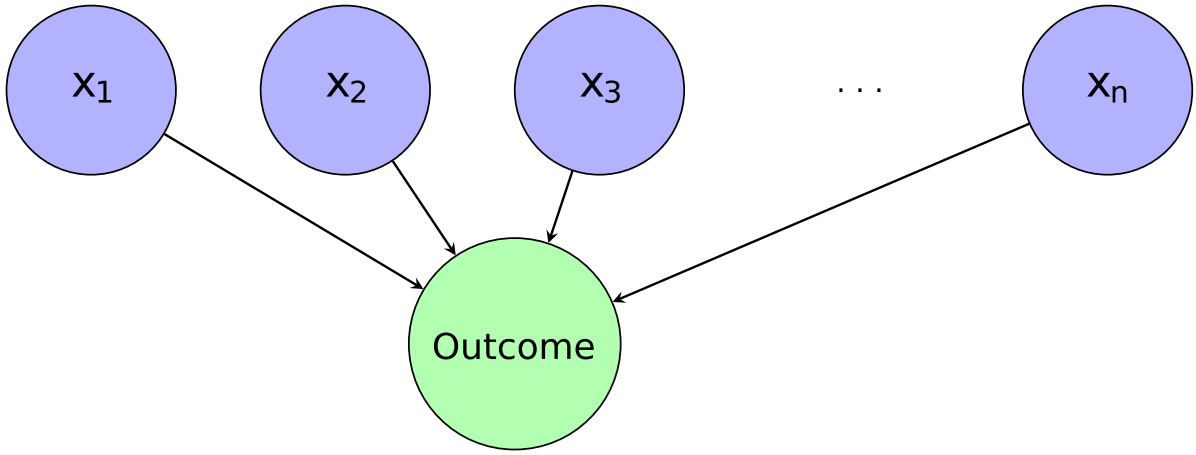

Figure 4.1: Relationship of input X with the outcome

Consequently, DiCE is used to facilitate end-user explainability. Its role is to extract explanations from an end-user perspective, ensuring they are accessible to non-expert stakeholders.

4.5 Proposed model

In summary, this study advanced a unified model that fuses explainable ML with PdM to (1) predict maintenance needs and (2) furnish explanations for the model predictions, as seen in Figure 4.2. Moreover, trained on a time series dataset containing maintenance logs, sensor data, and (a) machine and (b) water pump statuses, the proposed ML model predicts potential failures (prognostics), elucidating their underlying causes by component level (diagnostics) and addressing explainability through four key facets: (1) data, (2) model, (3) outcome, and (4) the end-user (Arrieta *et al.*, 2020; Doshi-Velez & Kim, 2017).

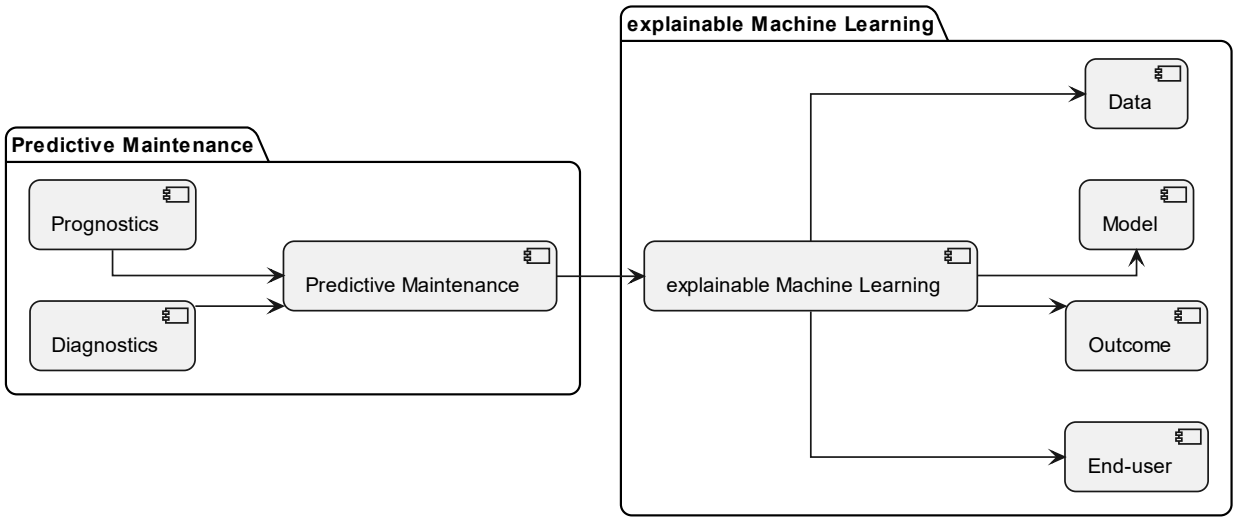

Figure 4.2: PdM and explainable ML illustration

4.6 Summary

This chapter discussed the application of explainable ML for PdM, aligning with the second part of the research aim: delivering explanations using explainable ML for PdM in SAF. Techniques for each explainability dimension were discussed. In the “data” dimension, the Deepchecks framework offers data expectations customised for expert stakeholders; this tool allows experts to understand the nuances of the data and aligns their expectations with the explanations derivable from the data. For the “model” and “outcome” dimensions, SHAP, a technique offering insights from both global and local standpoints, is again tailored for those with in-depth knowledge of the field; this approach helps dissect the model's decision-making process, offering clarity on how input data translates to specific outcomes. Addressing the “end-user” dimension, DiCE is geared toward simplifying explanations for those without technical expertise; this method bridges the gap between complex ML operations and practical understanding, making ML insights accessible to a broader audience. Next, the proposed model was illustrated to show the flow from PdM to explainable ML. Looking ahead, the ensuing chapter lays out the experimental design, a pivotal step in realising the research objectives.

CHAPTER 5 EXPERIMENTAL DESIGN

5.1 Overview

This study was determined to (1) predict maintenance needs and (2) provide explanations using explainable ML for PdM in SAF. In addition, this chapter presents the experimental design for the study, which encompasses the research paradigm, a supporting theoretical framework, the datasets used, ethical considerations, a research design paramount to the study, the configuration of the system, and the configuration of the parameters. To achieve its objectives, the study conducted various experiments.

5.2 Research paradigm

Theoretical considerations deserve attention to conceptualise the experimental design, namely, the research paradigm and the theoretical framework. The study implemented a quantitative methodology rooted in positivist ontology that underpins knowledge creation through measurements and systematic experiments (McKenna *et al.*, 2011; Turyahikayo, 2021). Furthermore, an objective epistemology made it possible to discern and quantify truth, ensuring that the study's conclusions remained independent of the researcher (Donald, 2018). Another essential feature of this study is the application of deductive reasoning because it fosters reliability measures (Goel & Dolan, 2003).

5.3 Theoretical framework

This study situates itself in the field of computer science (CS). Embracing a positivist ontology, as previously stated, the researcher employs Design Science Research (DSR) as its robust theoretical framework. DSR serves as a help and an engine propelling Computer Science projects to construct knowledge systematically (Hevner *et al.*, 2004). The researcher's choice to harness DSR stems not only from its comprehensive nature but also from its capacity to stimulate the development of artifacts. From intricate algorithms and sophisticated software to inventive approaches, ML models, and novel ideas, these artifacts bear witness to the effectiveness of DSR (Akoka *et al.*, 2023).

5.4 Data

In addition, this study used open-source (freely available to the public) data from the Internet to experiment and evaluate the proposed ML model, and foster reproducibility. The Telemetry for Predictive Maintenance (Microsoft, 2021) and the Pump Sensor datasets (Pump, 2019), with 11 and 54 unique attributes, helped test the proposed ML model.

5.5 Derived data

After pre-processing, the Telemetry for Predictive Maintenance dataset had 292019 observations with 19 unique attributes. In comparison, the Pump Sensor dataset comprised 218880 observations with 4 attributes. Both datasets revealed a significant class imbalance. In the Telemetry dataset, normal (class none) instances exceeded component failure instances (comp1, comp2, comp3, comp4) by 98% (Figure 5.1). Similarly, in the Pump Sensor dataset, normal status (class normal) instances surpassed abnormal (recovering, broken) by 93% (Figure 5.2). Thus, CSML was applied before the modelling stage to address these imbalances, ensuring fair representation and accuracy across different classes.

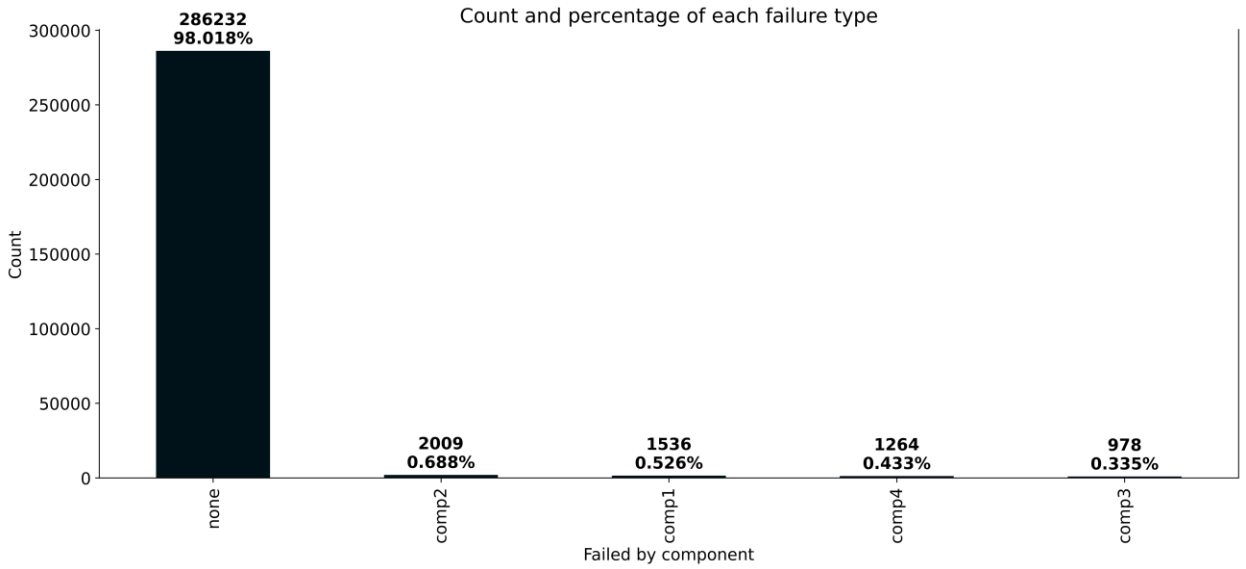

Figure 5.1: Count and percentage of each failure by component

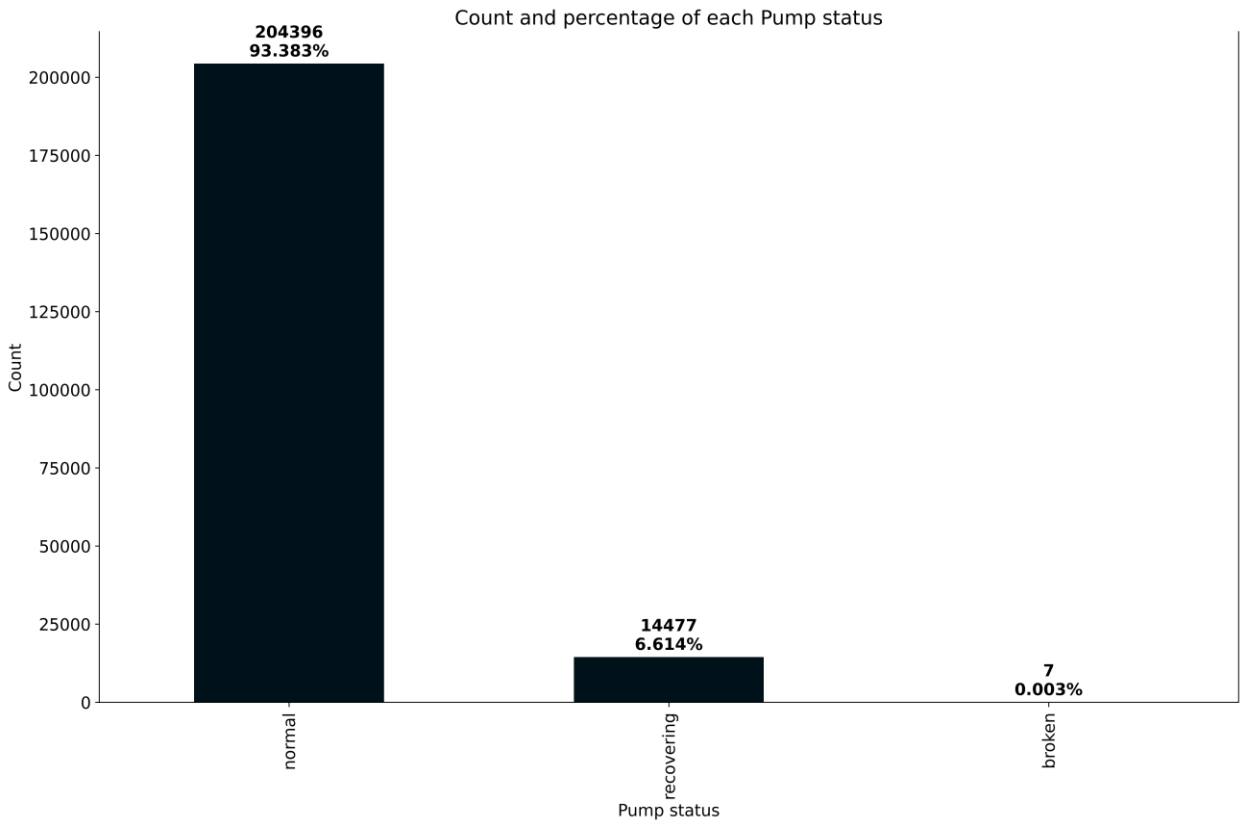

Figure 5.2: Count and percentage of each Pump status

5.6 Ethical considerations

The datasets used for this study were open-source and freely available. Moreover, the data did not contain personally identifiable information. Therefore, this study did not pose any risk to animals or humans.

5.7 Research design

The objectives informed the research design adopted for this study. Consequently, due to its research-centric and systematic nature, the research design embraced the Systematic Research on Big Data (SRBD) methodology (Das *et al.*, 2015). However, a review of "Data Science Approaches" asserted that SRBD fails to clarify roles and communication among team members throughout various project stages (Martinez *et al.*, 2021). Despite that, these factors did not affect this study, since it was conducted in an academic framework. Moreover, SRBD is data-driven and agile, which promotes reproducibility.

Furthermore, the standard SRBD methodology consists of 7 stages: information extraction and cleaning, preliminary data analysis, research goal, research data design, model and feature selection, output evaluation, and visualisation.

The proposed research design encompassed 9 stages, each addressing distinct research objectives to meet the aim of this study. Stage one, (1) data acquisition, was a fundamental resource for subsequent stages. Moreover, (2) explainable ML aimed to identify and explain the predictions made by the proposed ML model. These stages ensured comprehensive coverage of all research objectives, as shown in Figure 5.3.

Therefore, a meticulous SLR with bibliometric analysis was carried out, primarily addressing the research objective of (i) determining the current landscape of ML models and the necessity for explainable ML. The outcomes of stage 1 set the subsequent stages in motion. Stages 2 to 6 were crucial in tackling the research objective of (ii) developing an ML model to predict maintenance needs. Stage 7 evaluated the proposed ML model, squarely dealing with the research objective of (iii) model evaluation. The final stages, 8 and 9, were essential to address the research objective of (iv) identifying and providing explanations for the predictions made by the model.

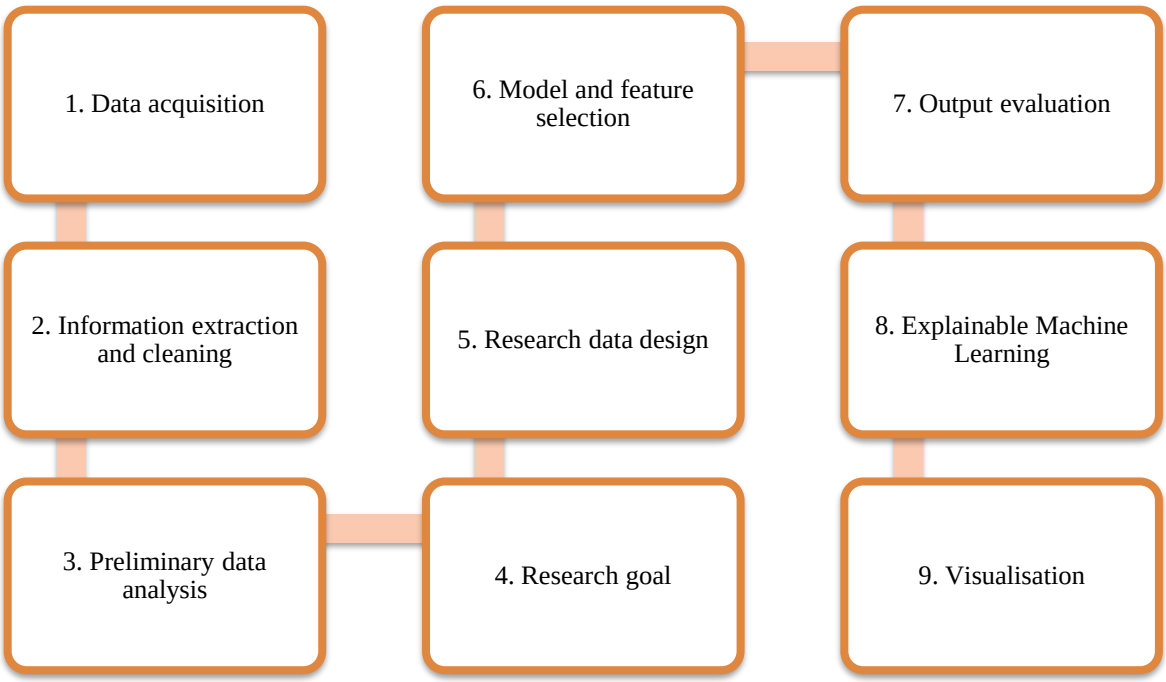

Figure 5.3: Modified SRBD stages

5.7.1 Data acquisition

Data acquisition propelled this study by garnering needed data; this study sourced publicly available data, one machine-focused and another centred around an irrigation system. Machine data depicted machine failure and was also used for root cause analysis, demonstrating component-specific failure. The irrigation system illustrated the pump failure.

5.7.2 Information extraction and cleaning

Information extraction and cleaning initiated data pre-processing, rendering the data fit for subsequent stages (Luengo *et al.*, 2020); this procedure encompassed (1) data cleaning, (2) data formatting, (3) feature engineering, and (4) data integration.

i. Data cleaning

Data cleaning required addressing absent values; this study used the "forward fill", also known as Last Observation Carried Forward (LOCF), which replaces missing values with their preceding present value. Moreover, this method is popular when working with time series data (Kamalov & Sulieman, 2021; Wijesekara & Liyanage, 2021). The “forward fill” method was selected over the alternative "backward fill" because it prevents data leakage. In particular, filling missing values with subsequent values proves unrealistic, particularly in time series data, given that these values have yet to materialise; this study employed the "forward fill" technique, honouring the temporal order of the dataset.

ii. Data formatting

This process involved formatting variables into fitting data types, namely categorical and numerical, thus enabling efficient processing in subsequent steps.

iii. Feature engineering

Feature engineering was thoroughly implemented, extracting additional information from the data and serving as additional predictive variables. Thus, the process unfolded in four steps (Figure 5.4): (1) DateTime features, (2) Lag features, (3) Window features, and (4) Periodic Cyclic features.

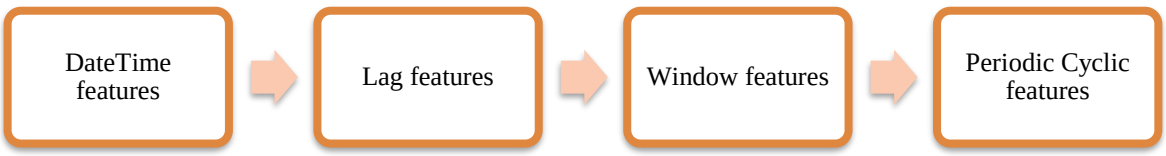

Figure 5.4: Four-step feature engineering

a. DateTime features

The features produced from the DateTime variables encapsulate data like "month", "week", "day_of_week", "day_of_month", "hour", and "weekend".

b. Lag features

These features encapsulate historical data, which are used to make predictions and make up lag features. The extraction of lag features drew inspiration from the data's inherent nature, including its collection rate. The identified lags bore the labels "t-1", "t-3", and "t-24", with “t” denoting the time passed.

c. Window features

The window features embodied mathematical calculations over an established span preceding the desired prediction time. The derived window features encapsulate the average from the preceding 3 hours in the time series, all to predict the forthcoming hour.

d. Periodic Cyclical features

Certain variables exhibit cyclical patterns, such as hours, days, or months. These cyclic variables can be transposed using a sine and cosine transformation relative to the variable’s cycle; this transformation brings the values of the variables, initially distant, closer to each other. For instance, the 3rd of October is closer to the 4th of November than the 7th of May; this relationship escapes the numerical representation of these variables. However, this could be altered by applying sine and cosine transformations to these variables.

Furthermore, observe that after feature engineering, variables are referred to as predictors or features, while the target variable garners the title of either outcome variable or target label.

iv. Data integration

This study required the merging of data, which was unique only to the Telemetry for Predictive Maintenance dataset (Microsoft, 2021), as it was split across multiple files. The merging was performed similarly to the merging of database tables using unique identifiers.

5.7.3 Preliminary data analysis

Assisted in understanding the data used while drawing initial insights. Grasping the data required focus on descriptive statistics, data types, missing values, and duration. Moreover, outlier detection was excluded, as robust ML algorithms applied in this study were resilient to outliers.

5.7.4 Research goal

Driven by identified research gaps, this study focused on (1) predicting maintenance needs and (2) providing explanations using explainable ML for PdM in SAF. Therefore, this study harnessed the potential of PdM and explainable ML to fulfil the research aim.

5.7.5 Research data design

The research data design structured the data so that it is suitable for creating the proposed ML model. Procedures such as encoding and maintaining temporal order during the train-test split played prominent roles, owing to the use of a time series dataset. The first step was to split the dataset into subsets - one for training and the other for testing the proposed ML model. Given the time series nature of the data, the splits required careful adherence to the temporal order, thereby excluding any shuffling or randomisation. The resulting splits, designed to retain temporal integrity, yielded a distribution of 70% for training purposes, with the remaining 30% allocated for testing.

Subsequently, the process involved pre-processing incorporating both encoding and scaling. Each split was individually processed, mitigating the potential for data leakage and the undesirable scenario where information seeps from the training set into the testing set, thus undermining the credibility of the research results. Encoding was applied to categorical features, enabling their efficient use during modelling. Crucially, the target label underwent separate encoding as another measure against data leakage. Using standardisation, numeric elements were scaled to ensure that no feature could suppress its counterparts.

5.7.6 Model and feature selection

i. Model selection

The models used in this study drew their foundation from a comprehensive review of related literature. Also, note that, given the widespread use of the test F1 score and for comparative analysis, this metric became the benchmark for identifying superior DL and ML algorithms in each dataset, which were then used for the explainability stage (Gashi *et al.*, 2023; Steurtewagen & Van den Poel, 2021).

ii. Feature selection

Choosing features that function as strong predictors of the target label was the primary task for this step; this study carried out feature selection in 4 steps (Figure 5.5): (1) variance thresholding, (2) pairwise correlation, (3) Recursive Feature Elimination with Cross-Validation (RFECV), and (4) Boruta-SHAP.

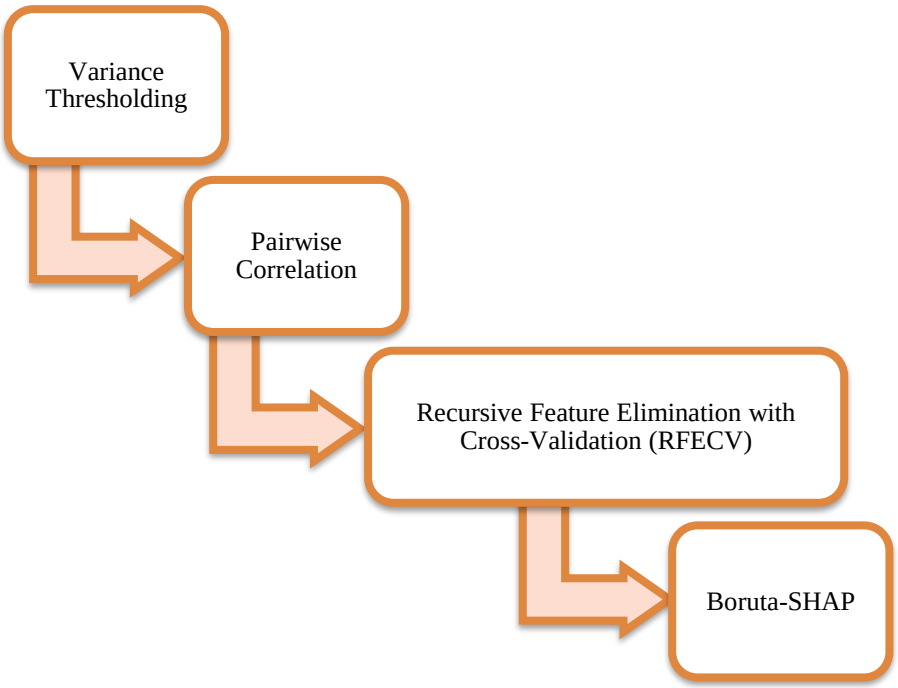

Figure 5.5: Four-step feature selection

a. Variance Thresholding

Variance Thresholding (VT) operates on the premise that features possessing low variance contribute insignificantly to predictions (Lane *et al.*, 2003). Such features, marked by scarce unique values or insufficiently low variances, bear little importance. VT serves as an effective tool for their elimination.

A pressing concern before implementing VT revolves around the proportionality of features. As the values for each feature increase, their variance increases exponentially. Consequently, features bearing dissimilar distributions manifest contrasting scales, rendering variance comparison insecure. Thus, demanding some measure of normalisation, ensuring that each feature adheres to an identical scale before applying VT.

Following normalisation (in this instance, each example is divided by its feature's average), one should select a threshold between 0 and 1. Rather than using the transform() approach of the VT estimator, the preference lies with get_support(), which provides a Boolean mask (True values for features that warrant retention); this mask subsets data while maintaining column names. Given its ability to handle such data, this method finds an application specifically with numerical features.

b. Pairwise Correlation

A metric showing linear association (Pearson's correlation coefficient) exists (Wilhelm, 2008). A correlation of 0.9 between features A and B means that the values of A can predict the values of B with an accuracy of 90%. Consequently, in a dataset containing A, there exists no necessity for B, or alternatively. Similarly, this approach finds application with numeric features.

c. Recursive Feature Elimination with Cross-Validation

Subsequently, the features were further filtered, guided by their influence on the effectiveness of the model; this concept systematically eliminates features, relying on cross-validation until the most minute feature set remains (Wang & Chen, 2019). The use of Linear Regression (LR), renowned for its inherent explainability, served as the model for RFECV; this algorithm was invoked exclusively on numeric features, given its compatibility with such data types.

d. Boruta-SHAP

Boruta-SHAP marries two algorithms, catering to both numerical and categorical data types. Boruta was developed as an envelope for the Random Forest classifier algorithm (Kursa *et al.*, 2010). Although it is an effective strategy for feature selection, Boruta heavily relies on feature importance calculations, which could be biased for the data.

In addition, Boruta may pose high computational costs, accompanied by inconsistency in feature ranking. Here, SHAP enters, incorporating SHAP as Boruta's feature selection method, giving birth to the Boruta-SHAP feature selection technique (Keany, 2023); this fusion provides additive feature explanations and benefits of the SHAP method while maintaining the robustness of Boruta to select only salient features. The algorithm that embodies Boruta-SHAP operates in this way:

i. Initiate by creating duplicates of every feature in the dataset. Then name them “shadow” and concatenate them with “feature_name”. Shuffle these recently added features, thus eliminating their associations with the target label.

ii. Invoke the Random Forest classifier on the extended data, incorporating random shadow features. Subsequently, prioritise features using the SHAP TreeSHAP explainer for efficient computation, surpassing the SHAP explainer; this further justifies using the Random Forest classifier during shadow feature creation.

iii. Formulate a threshold by harnessing the peak significance score derived from the shadow features. Subsequently, assign a hit to any features that have surpassed this threshold.

iv. Conduct a two-sided equality T-test for every unassigned feature.

v. Features that fall below the threshold bear “little significance”, warranting their removal. Conversely, consider those features surpassing the threshold as “highly significant”.

vi. Eliminate all shadow features, then proceed to replicate the procedure. Continue until each feature possesses an assigned significance or until the algorithm hits the previously defined limit for the Random Forest iterations.

5.7.7 Output evaluation

In this study, proximity to 1 means better performance for the model. Equally worthy of mention is the adoption of class weights, assigning a weight to a class proportionate to the observation count of the class (CSML); this method further limits the constraints posed by an imbalanced dataset.

Furthermore, this stage involved evaluating the performance of the proposed ML model; this study used three metrics (F1 score, accuracy, ROC-AUC), widely used in academic literature and those particularly tailored for dealing with imbalanced data (Carrington *et al.*, 2023; Gashi *et al.*, 2023; Steurtewagen & Van den Poel, 2021; Wang & Yao, 2013; Wu *et al.*, 2021). The ML algorithm, also called a classifier, could land within any category of the confusion matrix: *TP, TN, FP, or FN*. Where: TP denotes True Positive (correctly identified), TN denotes True Negative (correctly rejected), FP denotes False Positive (incorrectly identified), and FN denotes False Negative (incorrectly rejected). Moreover, the performance evaluation of a classification algorithm hinges on using the confusion matrix; this matrix presents correct classifications compared with erroneous ones, all categorised per class (Caelen, 2017; Minaee *et al.*, 2020). The integral components for its construction comprise True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) counts.

Furthermore, the F1 score embodies the harmonic average, combining recall and precision; the former gauges the accurate prediction of true positives, whilst the latter quantifies the misrepresentation of the positive class. A mathematical definition of the F1 score metric emerges in Equation (3). Note that the precision and recall must first be computed:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{2}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \tag{3}$$

In this context, the accuracy shows the proportion of accurate classifications relative to the total classifications. A mathematical definition of the accuracy metric emerges in Equation (4):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4}$$

The ROC visually shows the effectiveness of the model, while the AUC numerically ranks it from 0 to 1 (Hand & Till, 2001; McClish, 1989). The AUC's versatility in appraising performance across various thresholds renders it crucial, mainly when classes differ in significance or risk. This study used the Trapezoidal rule for AUC calculation, among other methods like Riemann Sums and Simpson's rule; this approach is particularly relevant here, as accurate failure prediction is deemed more critical than correctly identifying non-failures, reflecting the unequal importance of different classes in this context.

5.7.8 Explainable Machine Learning

Designed for transparency in predictions across four dimensions: (1) data, (2) model, (3) outcome, and (4) end-user (Arrieta *et al.*, 2020; Doshi-Velez & Kim, 2017).

5.7.9 Visualisation

SHAP plots and CFE tables are used for intuitive explanation presentation.

5.8 System and Parameter configuration

Research experiments were done using an Intel(R) Core (TM) i5.10500H Central Processing Unit (CPU) at 2.50 GHz speed, and 16174 MB RAM, running on Windows 11 Pro 64-bit Operating System (OS) with a 6009 MB Nvidia GeForce RTX 3060 laptop Graphics Processing Unit (GPU). The Python 3.8.17 language underpinned the programming. The study detailed configurations for top-performing DL and ML algorithms per dataset. It also outlined the parameter settings for the four explainability dimensions: (1) data, (2) model, (3) outcome, and (4) end-user (Arrieta *et al.*, 2020; Doshi-Velez & Kim, 2017). The sequence began with DL configurations for the Telemetry for Predictive Maintenance dataset, followed by those for explainability. It then presented DL configurations for the Pump Sensor dataset and their explainability aspects, with a similar pattern for ML configurations.

5.8.1 Telemetry for Predictive Maintenance dataset configurations

This study divided its dataset, allocating 70% to training and 30% to testing. A consistent random state, fixed at 777, was applied throughout all experiments. The model input comprised 19 features, with output targeting classification across 5 classes.

i. Deep Learning (Telemetry for Predictive Maintenance)

The Convolutional LSTM Neural Network's parameters, displayed in Figure 5.6, involve a sequence where input data are first transformed, normalised, and scaled; this ensures efficient training. The data are then reshaped into a (19, 1, 1) configuration, followed by a 1D Convolutional LSTM with four output channels. Subsequently, the input is flattened into a vector, retaining its initial form. To enhance regularisation, a certain percentage of inputs are randomly dropped. The final step involves a fully connected layer with 5 output units, facilitating the classification into 5 classes.

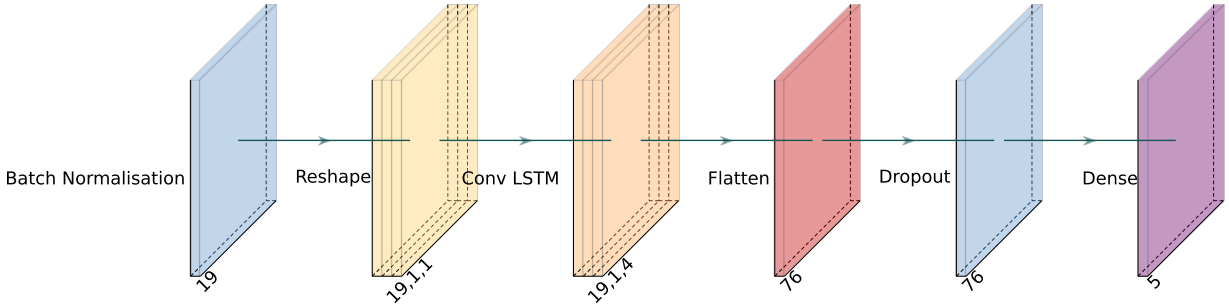

Figure 5.6: DL - Convolutional Long-Short-Term Memory Neural Network parameters (Telemetry for Predictive Maintenance)

a. Data explainability (Telemetry for Predictive Maintenance)

Table 5.1 presents Deepchecks data explanation parameters. These are the precise parameters used to extract insights, thus gauging the depth of explanations derivable from the data for the data dimension.

Table 5.1: Deepchecks Data explainability parameters (Telemetry for Predictive Maintenance)

Parameter	Value	Definition
n_samples	len(dataset)//2	The number of samples to use in the data integrity check
timeout	600	The maximum amount of time (in seconds) to allow for the check
n_top_columns	50	The number of top columns to include in the check
n_to_show	50	The number of results to show

b. Model explainability (Telemetry for Predictive Maintenance)

Table 5.2 shows the parameters of the DL SHAP model explainability, shedding light on the global behaviour of the model, for the model dimension; this offers insight into the variables steering the model's decision-making processes globally.

Table 5.2: DL - SHAP Model explainability parameters (Telemetry for Predictive Maintenance)

Parameter	Value	Definition
sample_size	25	Number of samples taken from the training data for SHAP analysis
X_sub	df_train	Subset of the training data (excluding the target variable) used for SHAP analysis
explainer	SHAP explainer object	An object that can calculate SHAP values, created using the model's prediction function and sampled data
shap_values_multiple	df_test[0:sample_size]	The SHAP values for the test data
class_indices	All five classes	List of class indices for which SHAP summary plots will be generated
class_label	[‘comp1’, ‘comp2’, ‘comp3’, ‘comp4’, ‘none’]	The label of the class corresponding to the current class index

c. Outcome explainability (Telemetry for Predictive Maintenance)

Table 5.3 unveils DL SHAP parameters, showing the model's local behaviour for the outcome dimension. Such insights navigate the model's decision rationale on an individual prediction basis.

Table 5.3: DL - SHAP Outcome explainability parameters (Telemetry for Predictive Maintenance)

Parameter	Value	Definition
sample_size	25	Number of samples taken from the training data for SHAP analysis
X_sub	df_train	Subset of the training data (excluding the target variable) used for SHAP analysis

explainer	SHAP explainer object	An object that can calculate SHAP values, created using the model's prediction function and sampled data
shap_values	df_test[0:1]	The SHAP values for the test data
class_indices	All five classes	List of class indices for which SHAP summary plots will be generated
class_label	[‘comp1’, ‘comp2’, ‘comp3’, ‘comp4’, ‘none’]	The label of the class corresponding to the current class index

d. End-user explainability (Telemetry for Predictive Maintenance)

Table 5.4 reveals DL DiCE parameters, generating "what if" scenarios as counterfactuals for the end-user dimension. These counterfactuals serve as guides, illuminating decisions users might make to achieve desired outcomes.

Table 5.4: DL - DiCE end-user explainability parameters (Telemetry for Predictive Maintenance)

Parameter	Value	Definition
data_object	dice_ml.Data object	DiCE data object prepared using the input dataset
backend	'TF'+tf.__version__[0]	TensorFlow backend version used (e.g., 'TF2' for TensorFlow version 2.x)
model_object	dice_ml.Model object	DiCE model object created using the TensorFlow model
explainer	dice_ml.Dice object	DiCE explanation object instantiated with the data object, model object, and the method set to 'random'
desired_classes	All other classes	List of desired classes for which counterfactuals are generated
test_query	DataFrame object	Subset of a dataset (excluding the target variable) representing a test query for which counterfactuals are generated
total_CFs	5	Total number of counterfactual instances to generate for each desired class
features_to_vary	‘all’	Specification of features to vary during counterfactual generation

proximity_weight	1.5	Weight is assigned to proximity in the counterfactual generation process. Feature-wise distance from the original input
diversity_weight	1.0	The weight assigned to diversity in the counterfactual generation process. Feature-wise distance between each counterfactual pair
stopping_threshold	0.5	Threshold for stopping the counterfactual generation process

ii. Machine Learning (Telemetry for Predictive Maintenance)

Bagging uses an ensemble of DTs (Figure 5.7). Weights find themselves assigned to classes, harmonising model performance. The entropy criterion comes into play for DT node splitting. The maximum depth of the DT receives a specification. The ensemble boasts 100 DT estimators. Processor cores maximised (-1) for parallel processing, accelerating model training. A random state, fixed at 777, persists, applying to all experimental procedures.

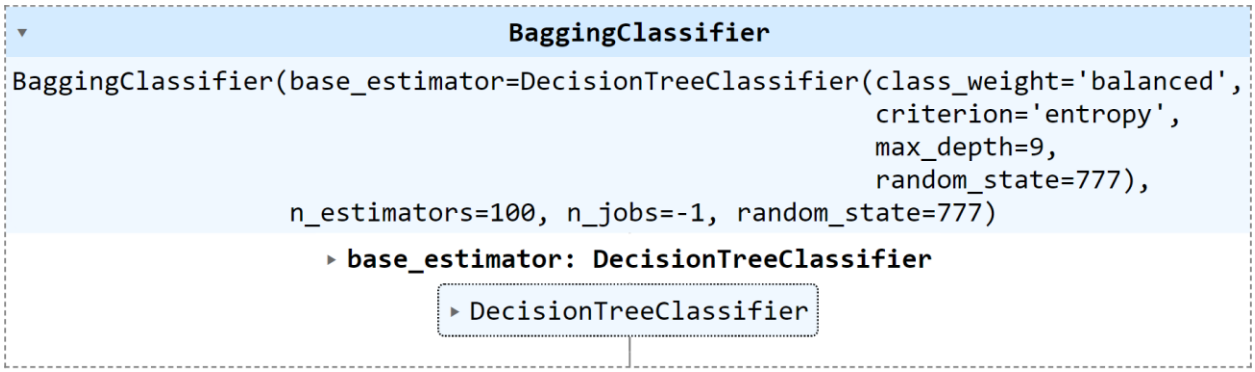

Figure 5.7: ML - Bagging classifier parameters (Telemetry for Predictive Maintenance)

a. Model explainability (Telemetry for Predictive Maintenance)

Table 5.5 shows the SHAP parameters for ML model explainability used to show the model's behaviour for the model dimension; this gives a deeper understanding of the variables that guide the model's global decision-making trajectory.

Table 5.5: ML - SHAP Model explainability parameters (Telemetry for Predictive Maintenance)

Parameter	Value	Definition
sample_size	100	Number of samples taken from the training data for SHAP analysis
X_sub	df_train	Subset of the training data (excluding the target variable) used for SHAP analysis
explainer	SHAP explainer object	An object that can calculate SHAP values, created using the model's prediction function and sampled data
shap_values_multiple	df_test[0:sample_size]	The SHAP values for the test data
class_indices	All five classes	List of class indices for which SHAP summary plots will be generated
class_label	[‘comp1’, ‘comp2’, ‘comp3’, ‘comp4’, ‘none’]	The label of the class corresponding to the current class index

b. Outcome explainability (Telemetry for Predictive Maintenance)

Table 5.6 reveals the ML SHAP parameters, showing the model's local behaviour for the outcome dimension; these insights show the model's decision-making pathways, focusing on individual predictions.

Table 5.6: ML - SHAP Outcome explainability parameters (Telemetry for Predictive Maintenance)

Parameter	Value	Definition
sample_size	100	Number of samples taken from the training data for SHAP analysis
X_sub	df_train	Subset of the training data (excluding the target variable) used for SHAP analysis
explainer	SHAP explainer object	An object that can calculate SHAP values, created using the model's prediction function and sampled data
shap_values	df_test[0:1]	The SHAP values for the test data
class_indices	All five classes	List of class indices for which SHAP summary plots will be generated

class_label	[‘comp1’, ‘comp2’, ‘comp3’, ‘comp4’, ‘none’]	The label of the class corresponding to the current class index

c. End-user explainability (Telemetry for Predictive Maintenance)

Table 5.7 shows the ML DiCE parameters to generate "what if" (counterfactual) scenarios for the end-user dimension; these guide users through decisions toward preferred prediction outcomes.

Table 5.7: ML - DiCE end-user explainability parameters (Telemetry for Predictive Maintenance)

Parameter	Value	Definition
data_object	dice_ml.Data object	DiCE data object prepared using the input dataset
backend	‘sklearn’	The backend used for the DiCE model (in this case, sklearn)
model_object	dice_ml.Model object	DiCE model object created using the sklearn model
explainer	dice_ml.Dice object	DiCE explanation object instantiated with the data object, model object, and the method set to 'random'
desired_classes	All other classes	List of desired classes for which counterfactuals are generated
test_query	DataFrame object	Subset of a dataset (excluding the target variable) representing a test query for which counterfactuals are generated
total_CFs	5	Total number of counterfactual instances to generate for each desired class
features_to_vary	‘all’	Specification of features to vary during counterfactual generation
proximity_weight	1.5	Weight is assigned to proximity in the counterfactual generation process. Feature-wise distance from the original input
diversity_weight	1.0	The weight assigned to diversity in the counterfactual generation process. Feature-wise distance between each counterfactual pair
stopping_threshold	0.5	Threshold for stopping the counterfactual generation process

5.8.2 Pump Sensor dataset configurations

The dataset was split, dedicating 70% for training and the rest for testing, ensuring uniformity in all trials with a random state of 777. It involved 4 distinct features as inputs and 3 classes as output for classification.

i. Deep Learning (Pump Sensor)

In Figure 5.8, the BiRNN and LSTM Neural Network parameters were meticulously set. The input data are first transformed, normalised, and scaled. The data are then reshaped into a (1, 4) configuration, followed by a BiRNN and LSTM with 16 output channels. To enhance regularisation, a certain percentage of inputs are randomly dropped. The final step involves a dense layer with 3 output units, facilitating the classification into 3 classes.

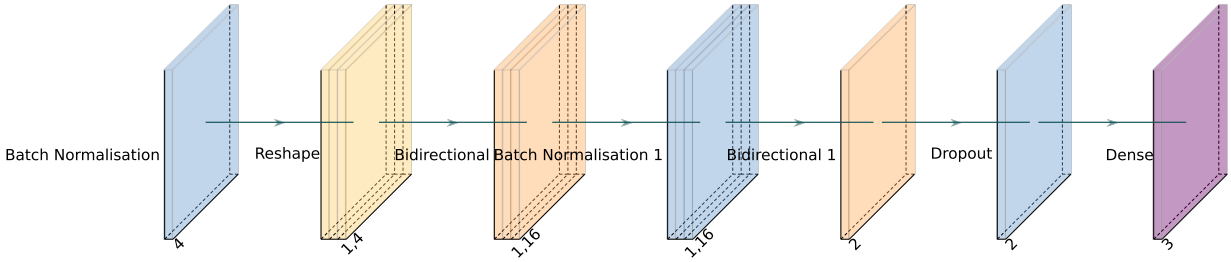

Figure 5.8: DL - Bidirectional Recurrent Neural Network and Long-Short-Term Memory Neural Network parameters (Pump Sensor)

a. Data explainability (Pump Sensor)

Table 5.8 shows the parameters for the Deepchecks data explanations. These are the precise parameters used to gauge the depth of explanations derivable from the data for the data dimension.

Table 5.8: Deepchecks Data explainability parameters parameters (Pump Sensor)

Parameter	Value	Definition
n_samples	len(dataset)//2	The number of samples to use in the data integrity check
timeout	600	The maximum amount of time (in seconds) to allow for the check
n_top_columns	50	The number of top columns to include in the check

n_to_show	50	The number of results to show

b. Model explainability (Pump Sensor)

Table 5.9 shows the parameters for the DL SHAP model explanations, shedding light on the global behaviour of the model, for the model dimension; this offers insight into the variables steering the model's decision-making processes globally.

Table 5.9: DL - SHAP Model explainability parameters (Pump Sensor)

Parameter	Value	Definition
sample_size	25	Number of samples taken from the training data for SHAP analysis
X_sub	df_train	Subset of the training data (excluding the target variable) used for SHAP analysis
explainer	SHAP explainer object	An object that can calculate SHAP values, created using the model's prediction function and sampled data
shap_values_multiple	df_test[0:sample_size]	The SHAP values for the test data
class_indices	All five classes	List of class indices for which SHAP summary plots will be generated
class_label	[‘comp1’, ‘comp2’, ‘comp3’, ‘comp4’, ‘none’]	The label of the class corresponding to the current class index

c. Outcome explainability (Pump Sensor)

Table 5.10 shows the parameters to explain the DL SHAP outcomes, showing the local behaviour of the model for the outcome dimension. Such insights navigate the model's decision rationale on an individual prediction basis.

Table 5.10: DL - SHAP Outcome explainability parameters (Pump Sensor)

Parameter	Value	Definition
sample_size	25	Number of samples taken from the training data for SHAP analysis
X_sub	df_train	Subset of the training data (excluding the target variable) used for SHAP analysis

explainer	SHAP explainer object	An object that can calculate SHAP values, created using the model's prediction function and sampled data
shap_values	df_test[0:1]	The SHAP values for the test data
class_indices	All five classes	List of class indices for which SHAP summary plots will be generated
class_label	[‘comp1’, ‘comp2’, ‘comp3’, ‘comp4’, ‘none’]	The label of the class corresponding to the current class index

d. End-user explainability (Pump Sensor)

Table 5.11 shows the DL DiCE parameters, generating "what if" scenarios as counterfactuals for the end-user dimension. These counterfactuals serve as guides, illuminating decisions users might make to achieve desired outcomes.

Table 5.11: DL - DiCE end-user explainability parameters (Pump Sensor)

Parameter	Value	Definition
data_object	dice_ml.Data object	DiCE data object prepared using the input dataset
backend	'TF'+tf.__version__[0]	TensorFlow backend version used (e.g., 'TF2' for TensorFlow version 2.x)
model_object	dice_ml.Model object	DiCE model object created using the TensorFlow model
explainer	dice_ml.Dice object	DiCE explanation object instantiated with the data object, model object, and the method set to 'random'
desired_classes	All other classes	List of desired classes for which counterfactuals are generated
test_query	DataFrame object	Subset of a dataset (excluding the target variable) representing a test query for which counterfactuals are generated
total_CFs	5	Total number of counterfactual instances to generate for each desired class
features_to_vary	‘all’	Specification of features to vary during counterfactual generation
proximity_weight	1.5	Weight is assigned to proximity in the counterfactual generation process. Feature-wise distance from the original input

diversity_weight	1.0	The weight assigned to diversity in the counterfactual generation process. Feature-wise distance between each counterfactual pair
stopping_threshold	0.5	Threshold for stopping the counterfactual generation process

ii. Machine Learning (Pump Sensor)

The parameter configurations for the Adaptive Boosting classifier are shown in (Figure 5.9). The learning rate is set to 0.2, dictating the step size for model updates. The Adaptive Boosting classifier includes 100 estimators (weak models) by definition. A random state, fixed at 777, persists, applying to all experimental procedures.

```
AdaBoostClassifier
AdaBoostClassifier(learning_rate=0.2, n_estimators=100, random_state=777)
```

Figure 5.9: ML - Adaptive Boosting classifier parameters (Pump Sensor)

a. Model explainability (Pump Sensor)

Table 5.12 shows the parameters for the ML SHAP model explanations, used to show the model's behaviour for the model dimension; this gives a deeper understanding of the variables that guide the model's global decision-making route.

Table 5.12: ML - SHAP Model explainability parameters (Pump Sensor)

Parameter	Value	Definition
sample_size	500	Number of samples taken from the training data for SHAP analysis
X_sub	df_train	Subset of the training data (excluding the target variable) used for SHAP analysis
explainer	SHAP explainer object	An object that can calculate SHAP values, created using the model's prediction function and sampled data
shap_values_multiple	df_test[0:sample_size]	The SHAP values for the test data

class_indices	All five classes	List of class indices for which SHAP summary plots will be generated
class_label	[‘comp1’, ‘comp2’, ‘comp3’, ‘comp4’, ‘none’]	The label of the class corresponding to the current class index

b. Outcome explainability (Pump Sensor)

Table 5.13 shows the parameters for explaining the ML SHAP outcomes, showing the model's local behaviour for the outcome dimension; these show the model's decision-making pathways, focusing on individual predictions.

Table 5.13: ML - SHAP Outcome explainability parameters (Pump Sensor)

Parameter	Value	Definition
sample_size	500	Number of samples taken from the training data for SHAP analysis
X_sub	df_train	Subset of the training data (excluding the target variable) used for SHAP analysis
explainer	SHAP explainer object	An object that can calculate SHAP values, created using the model's prediction function and sampled data
shap_values	df_test[0:1]	The SHAP values for the test data
class_indices	All five classes	List of class indices for which SHAP summary plots will be generated
class_label	[‘comp1’, ‘comp2’, ‘comp3’, ‘comp4’, ‘none’]	The label of the class corresponding to the current class index

c. End-user explainability (Pump Sensor)

Table 5.14 shows the parameters for the ML DiCE end-user dimension; these guide users through decisions towards preferred prediction outcomes.

Table 5.14: ML - DiCE end-user explainability parameters (Pump Sensor)

Parameter	Value	Definition
data_object	dice_ml.Data object	DiCE data object prepared using the input dataset

backend	‘sklearn’	The backend used for the DiCE model (in this case, sklearn)
model_object	dice_ml.Model object	DiCE model object created using the sklearn model
explainer	dice_ml.Dice object	DiCE explanation object instantiated with the data object, model object, and the method set to 'random'
desired_classes	All other classes	List of desired classes for which counterfactuals are generated
test_query	DataFrame object	Subset of a dataset (excluding the target variable) representing a test query for which counterfactuals are generated
total_CFs	5	Total number of counterfactual instances to generate for each desired class
features_to_vary	‘all’	Specification of features to vary during counterfactual generation
proximity_weight	1.5	Weight is assigned to proximity in the counterfactual generation process. Feature-wise distance from the original input
diversity_weight	1.0	The weight assigned to diversity in the counterfactual generation process. Feature-wise distance between each counterfactual pair
stopping_threshold	0.5	Threshold for stopping the counterfactual generation process

5.9 Summary

The chapter detailed the experimental design; this included the research paradigm, an underpinning theoretical framework, the datasets used, ethical considerations, a pivotal research design for the study’s research objectives, system configuration, and parameter configuration. In particular, the research design encompassed 9 stages: (1) data acquisition, (2) information extraction and cleaning, (3) preliminary data analysis, (4) research goal, (5) research data design, (6) model and feature selection, (7) output evaluation, (8) explainable Machine Learning, and (9) visualisation. The parameter configurations then detailed the (a) parameter, (b) value, and (c) definition, showing the exact configuration for each experiment to promote reproducibility. Consequently, the following chapter presents the results and discussion of this study.

CHAPTER 6 RESULTS AND DISCUSSION

6.1 Overview

This chapter presents the results of explainable ML and PdM in SAF in distinct sub-themes. The research presented discussions around prognostics and diagnostics for PdM. Prognostics predicts the conditions of the water pump, while diagnostics identify specific components of machine failure. Both use DL and ML algorithms. The research also considers the “data dimension”, assessing how data limitations and expectations influence insight extraction. In addition, this chapter presents global explainability, aiming to unravel the model's behaviour through model-centric approaches for the “model dimension”. Local explainability strives to explain single prediction instances, providing insights into the “outcome dimension”. Addressing the “end-user dimension” emphasises creating explanations that balance abstraction and detail. Furthermore, the discussions detail a comparative analysis with related studies, while following themes (prognostics, diagnostics, data, model, outcome, and end-user) and interpretation of the results, exposing implications and alignment with the related literature; this chapter concludes by summarising the results and discussion.

This research had a dual aim: (1) predict maintenance needs and (2) provide explanations using explainable ML for PdM in SAF. Chapter 2 conducted an SLR with bibliometric analysis, tackling the first research objective: (a) conducting an SLR using bibliometric analysis to determine the current landscape of ML models and the need for explainable ML. Chapter 5 addressed the following three objectives of this research to achieve its aim: (b) developing an ML model to predict maintenance needs, (c) evaluating the proposed ML model, and (d) then identifying and providing explanations for the predictions made by the proposed ML model.

6.2 Data analysis

Remember that the Telemetry for Predictive Maintenance data spans from 06:00:00 on the first day of 2015 to 06:00:00 on the first day of 2016, assembled every hour. Likewise, remember that Pump Sensor data extend from midnight on the first of April 2018 to one minute before midnight on the thirty-first of August the same year, with each minute accounted for and a total of 52 sensor units before pre-processing.

6.2.1 Attribute analysis

The attributes and definitions in the Telemetry for Predictive Maintenance dataset are shown in Table 6.1.

Table 6.1: Telemetry attributes

No	Attribute	Definition
1	datetime	When the reading was recorded
2	machineID	Unique ID for each machine
3	volt	The potential difference between two points in an electrical circuit
4	rotate	The speed at which the pump motor rotates
5	pressure	The effect at which water is carried inside the pipe
6	vibration	Rapid back-and-forth motion of the water pump
7	errorID	Five unique error types that a machine can have ['error1', 'error4', 'error3', 'error5', 'error2’]
8	comp	There are four types of components for which a machine goes for maintenance ['comp2', 'comp1', 'comp4', 'comp3']
9	model	Four unique machine models ['model3', 'model4', 'model2', 'model1']
10	age	Age of machine
11	failure	Failure due to a specific component ['comp1', 'comp3', 'comp4', 'comp2']

The attributes and meaning of the Pump Sensor dataset are shown in Table 6.2.

Table 6.2: Pump Sensor attributes

No	Attribute	Definition
1	datetime	When the reading was recorded
2	sensor_n	Specific sensor and reading (n = 52)
3	pump_status	Three pump states ['normal', 'broken', 'recovering']

6.2.2 Descriptive statistics

i. Telemetry for Predictive Maintenance

a. Telemetry

Delving into the descriptive statistics of the telemetry data, “count” denotes instance quantity, “mean” denotes average, “Standard Deviation” (STD) denotes data point dispersion, and “min” denotes minimal value. At the same time, “max” shows the maximum value in each column. Also note that "min" is shown as "light grey”, while "max" is denoted as "light blue”. The percentile representation is assigned to 25%, 50%, and 75%. Looking at Table 6.3, one notices the peak volt reaching 255.124710. The rotation speed clock is 695.020996, while the pressure measures 185.951996. Vibration hits 76.791069. A crucial observation regarding Telemetry for Predictive Maintenance data is the absence of missing values.

Machine variability: The “machineID” identifies data from 100 distinct machines; this points to a diverse set of machines. Analysing their behaviour yields insights into machine performance and variability. Voltage range: the “volt” exhibited a wide range, from about 97.33 to 255.12; this hints at the varying voltage levels in the machines. Monitoring voltage variations becomes crucial for stable machine operation. Rotational speed: “rotate” shows notable variation, ranging from 138.43 to 695.02; this signifies different rotational speeds in machines, affecting efficiency and wear and tear. Variations in pressure: “pressure” ranged from 51.24 to 185.95; this wide range showed varying pressure conditions between machines, which are vital for evaluating machine health and performance. Vibration levels: “vibration” ranges from 14.88 to 76.79. Elevated vibration levels may signal mechanical issues, underscoring the importance of monitoring vibrations for maintenance. Machine

count: “machineID” totals 876100, evenly distributed, showing that each machine contributes an almost equal number of data points; this balance in data distribution is essential for unbiased analysis and modelling. Consistent median: Across all attributes, the median values (50%) consistently cluster around the mean, showing relatively symmetric data distributions.

Table 6.3: Telemetry descriptive statistics

Attribute	count	mean	std	min	25%	50%	75%	max
machineID	876100.000000	50.500000	28.866087	1.000000	25.750000	50.500000	75.250000	100.000000
volt	876100.000000	170.777771	15.509114	97.333603	160.304928	170.607338	181.004490	255.124710
rotate	876100.000000	446.605164	52.673885	138.432068	412.305717	447.558151	482.176590	695.020996
pressure	876100.000000	100.858665	11.048679	51.237106	93.498182	100.425556	107.555229	185.951996
vibration	876100.000000	40.384998	5.370360	14.877054	36.777300	40.237247	43.784938	76.791069

b. Errors

The following vital observations became known when analysing the error data in Table 6.4. "Levels" provides a snapshot of values about a specific attribute. The "No. of Levels" quantifies the count of unique attribute values. In contrast, the "No. of Missing Values" enumerates the instances of incomplete data. “Datetime” attribute: The dataset boasts 2720 distinct timestamps and maintains a flawless record with no gaps; this robust and unbroken chronology supports its suitability for time-centric analysis. “ErrorID” attribute: The dataset features five unique error categories - error1, error2, error3, error4, and error5 - with no gaps in the data; this reveals that the dataset captures diverse error types, and none of these critical error categories are absent. “MachineID” attribute: In terms of machines, the dataset includes 100 unique machine IDs, each with complete data, ensuring comprehensive coverage and accountability for every machine.

Table 6.4: Errors descriptive statistics

Attribute	Levels	No. of Levels	No. of Missing Values	Percentage of Missing Values
datetime	[2015-01-01T06:00:00.000000000]	2720	0	0.0
machineID	[24, 73, 81, 43, 14, 76, 42, 72, 75, 97, 10]	100	0	0.0
errorID	['error1', 'error4', 'error3', 'error5', 'error2']	5	0	0.0

c. Maintenance

Analysing maintenance data revealed the presence of four distinct component types in Table 6.5. Moreover, the "datetime" attribute showed that data acquisition began on 1 June 2014 and ended on 16 June 2014, with 374 unique timestamps. The "datetime" attribute maintained a complete record without omissions, underscoring its suitability for time series analysis. The "comp" attribute signified four distinct component types (comp1, comp2, comp3, comp4). Analogous to the other attributes, it exhibited a flawless dataset without gaps, confirming the inclusion of records for each component type. "MachineID" attribute: Within this dataset reside precisely 100 distinct machine IDs, with no voids in the data; this signified the dataset's comprehensive representation of 100 distinct machines.

Table 6.5: Maintenance descriptive statistics

Attribute	Levels	No. of Levels	No. of Missing Values	Percentage of Missing Values
datetime	[2014-06-01T06:00:00.000000000]	374	0	0.0
machineID	[1, 6, 9, 11, 16, 18, 19, 20, 21, 26, 37, 39]	100	0	0.0

comp	['comp2', 'comp1', 'comp4', 'comp3']	4	0	0.0

d. Machines

For machine data analysis, notable observations emerged. The most senior machine in the dataset boasts an age of 20 years (Table 6.6). “Age” statistics: This dataset provides insight into machine ages. The average age of the machines is 11.33 years. A standard deviation of approximately 5.857 signifies variability in machine ages, with some significantly older and others notably younger. The age data range from 0 to 20 years, underscoring the diversity among machines. The quartile values, specifically the 25th, 50th, and 75th percentiles, shed light on the distribution of machine ages, with the median age (50%) resting at 12 years. “MachineID” statistics: This dataset encompasses a hundred machine IDs, signifying data collection from various machines. The mean “machineID” hovers at 50.5, denoting a near-even distribution of machine IDs. The standard deviation, around 29.011, reveals some variability in “machineID” values, though it remains within reasonable bounds. The machine IDs range from 1 to 100, affirming the comprehensive coverage of various machines in the dataset.

Table 6.6: Machine descriptive statistics

Attribute	count	mean	std	min	25%	50%	75%	max
machine ID	100.000000	50.500000	29.011492	1.000000	25.750000	50.500000	75.250000	100.000000
age	100.000000	11.330000	5.856974	0.000000	6.750000	12.000000	16.000000	20.000000

e. Failures

Looking at the failure data, a notable observation became apparent. Of a total of 100 machines, 98 machines encountered failures (Table 6.7). Also, the “datetime” attribute showed that 302 unique timestamps exist, and no values are absent; this attests to meticulous documentation of the time series data, encompassing a range of time points without any gaps. “MachineID” attribute: Ninety-

eight distinct machine IDs are present in the dataset, and no values are missing; this means that data from 98 distinct machines are contained in the dataset, with a comprehensive representation. The “failure” attribute delineates four distinct failure categories (comp1, comp2, comp3, comp4). Mirroring the other attributes, no values are absent; the dataset included records for each failure category, with none omitted.

Table 6.7: Failures descriptive statistics

Attribute	Levels	No. of Levels	No. of Missing Values	Percentage of Missing Values
datetime	[2015-01-02T03:00:00.000000000]	302	0	0.0
machineID	[16, 17, 22, 35, 45, 51, 56, 58, 59, 73, 79]	98	0	0.0
failure	['comp1', 'comp3', 'comp4', 'comp2']	4	0	0.0

i. Pump Sensor

Table 6.8 offered a comprehensive overview of the data, providing details on attributes, their respective levels, and the degree of absent data. The sensors garner various readings, with certain ones demonstrating a notable absence of values, demanding vigilant treatment through data imputation or during analysis. The attribute "Unnamed: 0" also served as an index or data identifier, covering values from 0 to 220320, without absent data. The "datetime" attribute denotes the data timestamp, formatted with date and time. It encompasses 220320 data points, all complete without any omissions. Moving on to "sensor_00" through "sensor_51”, these attributes denote various sensor readings, each with a distinct value range. For instance, "sensor_00" boasts 1254 unique levels. However, it presents 10208 absent values, constituting approximately 4.63% of the dataset. "Pump_status”, a categorical attribute, denotes the pump condition, featuring three levels: "normal”, "broken”, and "recovering”; this attribute retains a complete record with no missing data.

The spotlight falls on "sensor_15" and "sensor_50" for their notably high proportion of missing values. "Sensor_15" contains 100% missing values, rendering it devoid of data. Meanwhile, "sensor_50" exhibited a substantial 34.96% absence of values. Consequently, these attributes - "sensor_15", "sensor_50", and "Unnamed 0" - are shaded in light grey, signifying their exclusion from subsequent analysis.

Table 6.8: Pump Sensor descriptive statistics

Attribute	Levels	No. of Levels	No. of Missing Values	Percentage of Missing Values
Unnamed: 0	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]	220320	0	0.000000
datetime	[2018-04-01T00:00:00.000000000]	220320	0	0.000000
sensor_00	[2.465394, 2.444734, 2.460474, 2.445718]	1254	10208	4.633261
sensor_01	[47.09201, 47.35243, 47.13541, 47.04861]	832	369	0.167484
sensor_02	[53.2118, 53.1684, 53.168399810790994]	832	19	0.008624
sensor_03	[46.31076, 46.39757, 46.3975677490234]	589	19	0.008624
sensor_04	[634.375, 638.8889, 628.125, 636.4583]	7845	19	0.008624
sensor_05	[76.45975, 73.54598, 76.98898'	190752	19	0.008624
sensor_06	[13.41146, 13.32465, 13.317420000000002]	813	4798	2.177741
sensor_07	[16.13136, 16.037329999999997, 16.24711]	532	5451	2.474129
sensor_08	[15.56713, 15.617770000000002, 15.69734]	627	5107	2.317992
sensor_09	[15.05353, 15.010129999999998, 15.08247]	566	4595	2.085603
sensor_10	[37.2274, 37.86777, 38.57977, 39.48939]	198805	19	0.008624

sensor_11	[47.52422, 48.17723, 48.65607, 49.06298]	196369	19	0.008624
sensor_12	[31.11716, 32.08894, 31.67221, 31.95202]	187146	19	0.008624
sensor_13	[1.681353, 1.7084740000000005, 1.579427]	191984	19	0.008624
sensor_14	[419.5747, 420.848, 420.7494, 419.8926]	94565	21	0.009532
sensor_15	[nan]	1	220320	100.000000
sensor_16	[461.8781, 462.7798, 462.898, 461.4906]	110523	31	0.014070
sensor_17	[466.3284, 459.6364, 460.8858, 468.2206]	148001	46	0.020879
sensor_18	[2.565284, 2.500062, 2.509521, 2.604785]	152604	46	0.020879
sensor_19	[665.3993, 666.2234, 666.0114, 663.2111]	100423	16	0.007262
sensor_20	[398.9862, 399.9418, 399.1046, 400.5426]	92130	16	0.007262
sensor_21	[880.0001, 880.4237, 878.8917, 882.5874]	131084	16	0.007262
sensor_22	[498.8926, 501.3617, 499.043, 498.5383]	126402	41	0.018609
sensor_23	[975.9409, 982.7342, 977.752, 979.5755]	119287	16	0.007262
sensor_24	[627.674, 631.1326, 625.4076, 627.183]	133779	16	0.007262
sensor_25	[741.7151, 740.8031, 739.2722, 737.6033]	166000	36	0.016340
sensor_26	[848.0708, 849.8997, 847.7579, 846.9182]	179719	20	0.009078
sensor_27	[429.0377, 454.239, 474.8731, 408.8159]	203199	16	0.007262
sensor_28	[785.1935, 778.5734, 779.5091, 785.2307]	189279	16	0.007262
sensor_29	[684.9443, 715.6266, 690.4011, 704.6937]	201909	72	0.032680
sensor_30	[594.4445, 661.574, 686.1111, 631.4814]	2071	261	0.118464

sensor_31	[682.8125, 721.875, 754.6875, 766.1458]	2686	16	0.007262
sensor_32	[680.4416, 694.7721, 683.3831, 702.4431]	205415	68	0.030864
sensor_33	[433.7037, 441.2635, 446.2493, 433.9081]	200856	16	0.007262
sensor_34	[171.9375, 169.982, 166.4987, 164.7498]	203896	16	0.007262
sensor_35	[341.9039, 343.1955, 343.9586, 339.963]	201782	16	0.007262
sensor_36	[195.0655, 200.9694, 193.1689, 193.877]	201438	16	0.007262
sensor_37	[90.32386, 93.90508, 101.0406, 101.7038]	188899	16	0.007262
sensor_38	[40.36458, 41.40625, 41.92708, 42.70833]	603	27	0.012255
sensor_39	[31.51042, 31.25, 31.77083, 32.29166]	910	27	0.012255
sensor_40	[70.57291, 69.53125, 72.13541, 76.82291]	879	27	0.012255
sensor_41	[30.98958, 30.46875]	595	27	0.012255
sensor_42	[31.770832061767603, 31.77083]	700	27	0.012255
sensor_43	[41.92708, 41.66666, 40.88541, 41.40625]	714	27	0.012255
sensor_44	[39.6412, 39.3518524169922, 39.0625]	629	27	0.012255
sensor_45	[65.68287, 65.39352, 64.81481, 65.10416]	676	27	0.012255
sensor_46	[50.92593, 51.21528, 51.79398, 50.63657]	846	27	0.012255
sensor_47	[38.19444, 38.1944427490234, 38.77315]	622	27	0.012255
sensor_48	[157.9861, 155.9606, 158.2755, 164.6412]	1441	27	0.012255
sensor_49	[67.70834, 67.12963, 66.84028]	828	27	0.012255
sensor_50	[243.0556, 241.3194, 240.4514]	1136	77017	34.956881

sensor_51	[201.3889, 203.7037, 203.125]	1109	15383	6.982117
pump_status	[normal, broken, recovering]	3	0	0.000000

When analysing Pump Sensor data, one notices a pertinent statistical insight encompassing minimum, maximum, mean, and standard deviation. In Table 6.9, "min" is represented as "light grey", while "max" is represented as "light blue". The maximum pump speed is pushed to 2000, originating from "sensor_27". "Sensor_13": exhibited a notably high standard deviation compared to its mean and showed substantial data variability. The minimum figure registered at 0, but the maximum jumped to 31.188, hinting that this sensor may occasionally capture extreme values. "Sensor_14": displayed a relatively elevated mean of 376.860 alongside a standard deviation of 113.206; this shows that the data recorded by this sensor could encompass a broader spectrum and increased variability. "Sensor_16" and "sensor_17": Both exhibited notably elevated maximum values in contrast to their sensor counterparts, suggesting the possible presence of outliers or extreme data points in their measurements.

"Sensor_31" and "sensor_32": These sensors manifest elevated mean values (863.323 and 804.284, respectively) alongside relatively high standard deviations, signifying a more expansive data range; this observation could be significant for the current research's application. "Sensor_36": Exhibited a substantial standard deviation, signifying considerable data variation. It peaked at 984.061, hinting at data points considerably above the mean. "Sensor_38" to "sensor_44": These sensors have notably elevated maximum values compared to their means, indicating the existence of potential outliers or extreme values in their recorded measurements. "Sensor_48": Displayed a substantial standard deviation and a considerable span between minimum and maximum values. It showed a pronounced variability in the data recorded by this sensor. "Sensor_51": Featured a striking maximum value of 1000, significantly exceeding its mean; this underscores the presence of extreme values in the dataset. It should be noted that the DL and ML algorithms employed in this research demonstrate resilience in the face of outliers.

Table 6.9: Additional Pump Sensor descriptive statistics

Attribute	count	mean	std	min	25%	50%	75%	max
sensor_00	210112.000000	2.372221	0.412227	0.000000	2.438831	2.456539	2.499826	2.549016
sensor_01	219951.000000	47.591611	3.296666	0.000000	46.310760	48.133678	49.479160	56.727430

Attribute	count	mean	std	min	25%	50%	75%	max
sensor_02	220301.000000	50.867392	3.666820	33.159720	50.390620	51.649300	52.777770	56.032990
sensor_03	220301.000000	43.752481	2.418887	31.640620	42.838539	44.227428	45.312500	48.220490
sensor_04	220301.000000	590.673936	144.023912	2.798032	626.620400	632.638916	637.615723	800.000000
sensor_05	220301.000000	73.396414	17.298247	0.000000	69.976260	75.576790	80.912150	99.999880
sensor_06	215522.000000	13.501537	2.163736	0.014468	13.346350	13.642940	14.539930	22.251160
sensor_07	214869.000000	15.843152	2.201155	0.000000	15.907120	16.167530	16.427950	23.596640
sensor_08	215213.000000	15.200721	2.037390	0.028935	15.183740	15.494790	15.697340	24.348960
sensor_09	215725.000000	14.799210	2.091963	0.000000	15.053530	15.082470	15.118630	25.000000
sensor_10	220301.000000	41.470339	12.093519	0.000000	40.705260	44.291340	47.463760	76.106860
sensor_11	220301.000000	41.918319	13.056425	0.000000	38.856420	45.363140	49.656540	60.000000
sensor_12	220301.000000	29.136975	10.113935	0.000000	28.686810	32.515830	34.939730	45.000000
sensor_13	220301.000000	7.078858	6.901755	0.000000	1.538516	2.929809	12.859520	31.187550
sensor_14	220299.000000	376.860041	113.206382	32.409550	418.103250	420.106200	420.997100	500.000000
sensor_16	220289.000000	416.472892	126.072642	0.000000	459.453400	462.856100	464.302700	739.741500
sensor_17	220274.000000	421.127517	129.156175	0.000000	454.138825	462.020250	466.857075	599.999939
sensor_18	220274.000000	2.303785	0.765883	0.000000	2.447542	2.533704	2.587682	4.873250
sensor_19	220304.000000	590.829775	199.345820	0.000000	662.768975	665.672400	667.146700	878.917900
sensor_20	220304.000000	360.805165	101.974118	0.000000	398.021500	399.367000	400.088400	448.907900
sensor_21	220304.000000	796.225942	226.679317	95.527660	875.464400	879.697600	882.129900	1107.526000
sensor_22	220279.000000	459.792815	154.528337	0.000000	478.962600	531.855900	534.254850	594.061100
sensor_23	220304.000000	922.609264	291.835280	0.000000	950.922400	981.925000	1090.808000	1227.564000
sensor_24	220304.000000	556.235397	182.297979	0.000000	601.151050	625.873500	628.607725	1000.000000
sensor_25	220284.000000	649.144799	220.865166	0.000000	693.957800	740.203500	750.357125	839.575000

Attribute	count	mean	std	min	25%	50%	75%	max
sensor_26	220300.000000	786.411781	246.663608	43.154790	790.489575	861.869600	919.104775	1214.420000
sensor_27	220304.000000	501.506589	169.823173	0.000000	448.297950	494.468450	536.274550	2000.000000
sensor_28	220304.000000	851.690339	313.074032	4.319347	782.682625	967.279850	1043.976500	1841.146000
sensor_29	220248.000000	576.195305	225.764091	0.636574	518.947225	564.872500	744.021475	1466.281000
sensor_30	220059.000000	614.596442	195.726872	0.000000	627.777800	668.981400	697.222200	1600.000000
sensor_31	220304.000000	863.323100	283.544760	23.958330	839.062400	917.708300	981.249900	1800.000000
sensor_32	220252.000000	804.283915	260.602361	0.240716	760.607475	878.850750	943.877625	1839.211000
sensor_33	220304.000000	486.405980	150.751836	6.460602	489.761075	512.271750	555.163225	1578.600000
sensor_34	220304.000000	234.971776	88.376065	54.882370	172.486300	226.356050	316.844950	425.549800
sensor_35	220304.000000	427.129817	141.772519	0.000000	353.176625	473.349350	528.891025	694.479126
sensor_36	220304.000000	593.033876	289.385511	2.260970	288.547575	709.668050	837.333025	984.060700
sensor_37	220304.000000	60.787360	37.604883	0.000000	28.799220	64.295485	90.821928	174.901200
sensor_38	220293.000000	49.655946	10.540397	24.479166	45.572910	49.479160	53.645830	417.708300
sensor_39	220293.000000	36.610444	15.613723	19.270830	32.552080	35.416660	39.062500	547.916600
sensor_40	220293.000000	68.844530	21.371139	23.437500	57.812500	66.406250	77.864580	512.760400
sensor_41	220293.000000	35.365126	7.898665	20.833330	32.552080	34.895832	37.760410	420.312500
sensor_42	220293.000000	35.453455	10.259521	22.135416	32.812500	35.156250	36.979164	374.218800
sensor_43	220293.000000	43.879591	11.044404	24.479166	39.583330	42.968750	46.614580	408.593700
sensor_44	220293.000000	42.656877	11.576355	25.752316	36.747684	40.509260	45.138890	1000.000000
sensor_45	220293.000000	43.094984	12.837520	26.331018	36.747684	40.219910	44.849540	320.312500
sensor_46	220293.000000	48.018585	15.641284	26.331018	40.509258	44.849540	51.215280	370.370400
sensor_47	220293.000000	44.340903	10.442437	27.199070	39.062500	42.534720	46.585650	303.530100
sensor_48	220293.000000	150.889044	82.244957	26.331018	83.912030	138.020800	208.333300	561.632000

Attribute	count	mean	std	min	25%	50%	75%	max
sensor_49	220293.000000	57.119968	19.143598	26.620370	47.743060	52.662040	60.763890	464.409700
sensor_51	204937.000000	202.699667	109.588607	27.777779	179.108800	197.338000	216.724500	1000.000000

6.3 Experimental results

The experimental results reveal insights from the PdM alongside the explainable ML experiments. First, it dissects the results of both the employed DL and ML models. Next, it shifts towards explainable ML, which helps explain the four explainability dimensions drawn from scholarly literature: (1) data, (2) model, (3) outcome, and (4) end-user.

6.3.1 Predictive Maintenance results

In Chapter 2, the research navigated prognostics and diagnostics for PdM in SAF, focusing on system failure and component failure prediction literature. The Telemetry for the Predictive Maintenance dataset, initially with 11 attributes, expanded to 40 through feature engineering and refined to 19 for optimised performance, dismissing non-essential attributes like "datetime", "machineID", and "model", due to PdM in this research being a classification problem and not a forecasting one. Similarly, the Pump Sensor dataset, starting with 54 attributes, was distilled to 212, then to 4 after releasing irrelevant or incomplete attributes. With high test F1 scores (above 92%), accuracies (above 90%), and ROC-AUCs (above 80%), the models in this research exhibit strong adaptability to new data, avoiding overfitting. The DT and CatBoost classifiers, although slightly trailing, still deliver competitive results. These results, detailed in Tables 6.10 to 6.13, underscore the efficacy of the proposed model as a leading solution for PdM in SAF.

i. Deep Learning (Predictive Maintenance results)

In an accurate prognostics prediction endeavour on Telemetry for Predictive Maintenance dataset employing DL, as shown in Table 6.10, the metrics reveal various DL classifiers' prowess on training and test datasets. The "Overfitting" attribute, denoted as "False", shows the absence of overfitting. The Convolutional LSTM Neural Net reigned supreme in terms of test F1 score (0.970003) and test accuracy (0.942831), whilst the Convolutional Neural Net triumphed for test ROC-AUC (0.908591). Crucial findings include: the Convolutional LSTM Neural Net achieved the highest test F1 score

(0.970003), showing robust predictive performance on unseen data. All classifiers exhibited consistent train and test accuracy, showing robust generalisation on test data without noticeable overfitting. Although yielding slightly lower F1 scores, the Artificial Neural Net performed admirably, avoiding overfitting. Despite having the lowest test F1 score (0.939713), the Convolutional Neural Net delivered a commendable performance, successfully evading overfitting.

Table 6.10: DL - classifier comparison (Telemetry for Predictive Maintenance)

Classifier	Train F1 score	Test F1 score	Overfitting	Train accuracy	Test accuracy	Test ROC-AUC
Convolutional LSTM Neural Net	0.958015	**0.970003**	False	0.942831	**0.942831**	0.967616
BiRNN LSTM Neural Net	0.95516	0.963522	False	0.945424	0.945424	0.957594
Artificial Neural Net	0.945871	0.95583	False	0.922676	0.922676	0.939171
Convolutional Neural Net	0.936568	0.939713	False	0.905862	0.905862	**0.908591**

Inspecting Table 6.11 revealed the BiRNN LSTM Neural Net emerged as superior concerning test F1 score (0.998576) when performing predictive diagnostics on Pump Sensor data using DL. Similarly, the Artificial Neural Net asserted supremacy with unmatched test accuracy (0.998843). At the same time, the Convolutional LSTM Neural Net triumphed with unparalleled test ROC-AUC (0.935467). Robust performance characterises the four classifiers on training and test datasets, with elevated F1 scores and accuracy underscoring their proficiency in categorising test data points. The absence of overfitting, a phenomenon where models excel on training data but falter on test sets, is evident, with the "Overfitting" attribute consistently registering as "False". The BiRNN LSTM Neural Net and Artificial Neural Net outperform the rest, achieving the highest test F1 scores and test accuracies, highlighting their exceptional performance in predicting unseen data. The Convolutional Neural Net and LSTM Neural Net also demonstrate formidable test performance, although marginally trailing BiRNN and Artificial Neural Nets. Test ROC-AUC values provide an additional performance metric, and all classifiers post notably high ROC-AUC values, further emphasising their ability to discern unseen data points.

Table 6.11: DL - classifier comparison (Pump Sensor)

Classifier	Train F1 score	Test F1 score	Overfitting	Train accuracy	Test accuracy	Test ROC-AUC
BiRNN LSTM Neural Net	0.862548	**0.998576**	False	0.90576	0.998782	0.928758
Artificial Neural Net	0.861264	0.998264	False	0.905963	**0.998843**	0.802616
Convolutional Neural Net	0.948287	0.997471	False	0.924936	0.996573	0.800625
Convolutional LSTM Neural Net	0.955991	0.960525	False	0.933075	0.92518	**0.935467**

ii. Machine Learning (Predictive Maintenance results)

For prognostics prediction on Telemetry for Predictive Maintenance dataset using ML, as depicted in Table 6.12, the Bagging classifier reigns supreme for test F1 score (0.994729) and test accuracy (0.992466). In contrast, the CatBoost classifier triumphs with the highest test ROC-AUC (0.980435). Each classifier showed formidable efficacy, displaying impressive F1 scores, accuracies, and ROC-AUCs on training and test datasets. The absence of overfitting, as indicated by "Overfitting" tagged as "False", shows proficient generalisation capabilities towards unseen data. The Bagging classifier, LGBM classifier, DT classifier, and XGB classifier warrant attention due to their exceptional F1 scores and accuracy on test data, marking them as formidable contenders for classification tasks. Despite Random Forest, CatBoost, and AdaBoost classifiers displaying slightly lower F1 scores and accuracy on test data, their performance remains praiseworthy, successfully circumventing overfitting. ROC-AUC values for all classifiers nearing the digit 1 denote notable discrimination proficiency in classifying unseen data points.

Table 6.12: ML - classifier comparison (Telemetry for Predictive Maintenance)

Classifier	Train F1 score	Test F1 score	Overfitting	Train accuracy	Test accuracy	Test ROC-AUC
Bagging classifier	0.993066	**0.994729**	False	0.992466	**0.992466**	0.994373
LGBM classifier	0.993131	0.99447	False	0.992275	0.992275	0.993802

Decision Tree classifier	0.993036	0.994258	False	0.992432	0.992432	0.993836
XGB classifier	0.993043	0.993304	False	0.993733	0.993733	0.993779
RandomForest classifier	0.985309	0.985562	False	0.982755	0.982755	0.982501
CatBoost classifier	0.983746	0.984394	False	0.98028	0.98028	**0.980435**
AdaBoost classifier	0.969947	0.97304	False	0.977849	0.977849	0.981074

Table 6.13 revealed that the AdaBoost classifier reigned supreme across all three metrics: test F1 score, test accuracy, and test ROC-AUC when performing diagnostics prediction on the Pump Sensor dataset using ML. Among classifiers (AdaBoost, Random Forest, LGBM, XGB, Decision Tree, Bagging), "False" appears under the "Overfitting" attribute, showing their resilience against overfitting during training, thus showing robust generalisation capabilities for unseen data. The AdaBoost, Random Forest, LGBM, and XGB classifiers demonstrate admirable F1 scores on test data, highlighting their proficiency in predicting uncharted data, an essential characteristic for practical applications. The Random Forest classifier excels with an almost perfect test F1 score and high test accuracy, marking its position as a robust and efficient model. In contrast, the CatBoost classifier, with the lowest F1 scores and accuracy in both training and test datasets, suggests that it may not be an optimal choice among listed classifiers.

Table 6.13: ML - classifier comparison (Pump Sensor)

Classifier	Train F1 score	Test F1 score	Overfitting	Train accuracy	Test accuracy	Test ROC-AUC
AdaBoost classifier	0.945365	**0.999525**	False	0.951154	**0.999574**	**0.969477**
RandomForest classifier	0.99794	0.998902	False	0.997931	0.998766	0.936051
LGBM classifier	0.996472	0.998612	False	0.996417	0.998919	0.905535
XGB classifier	0.995269	0.998347	False	0.995255	0.998051	0.906245
Decision Tree classifier	0.971186	0.992829	False	0.969703	0.98803	0.494588

Bagging classifier	0.971175	0.992821	False	0.96969	0.988015	0.832808
CatBoost classifier	0.886011	0.951349	False	0.838352	0.90922	0.492795

Consequently, as established earlier, applying the test F1 score for comparative analysis, calls for its use in selecting distinguished DL and ML algorithms per dataset (Gashi *et al.*, 2023; Steurtewagen & Van den Poel, 2021). Therefore, the Convolutional LSTM Neural Network, BiRNN LSTM Neural Network, Bagging classifier, and AdaBoost classifier algorithms were used for the explainability stage.

6.3.2 Explainable Machine Learning results

Chapter 5's "Derived data" segment, showed two datasets: Telemetry for Predictive Maintenance and Pump Sensor, each with 5 and 3 classes, respectively. The analysis considers these class structures unless otherwise indicated. Key variables like "age", "datetime_day_of_month", "datetime_hour_sin", and "datetime_week" transformed into float values post-standardisation; this section presents and discusses explainability dimensions, using DL and ML algorithms as inputs.

i. Insights from the data

Using data-centric methods, the Telemetry for Predictive Maintenance dataset featured 21 columns, surpassing the original 19. In contrast, the Pump Sensor dataset had 6, exceeding the original 4. However, these extra columns, included categorical features and target variables, which were essential metadata, influencing neither data purity checks nor overall test outcomes. The research extracted deep insights directly from the data, assessing its purity and quality. The results showed meticulous preparation of the dataset, adhering to consistency, integrity, and predictability criteria.

a. Data results

In Table 6.14, the "Single Value in Column" check passed, affirming more than one unique value in all relevant columns. Similarly, the "Special Characters" check also passed, showing that the ratio of samples containing only special characters remains within the acceptable threshold of 0.1% across 21 relevant columns. The "Mixed Nulls" check likewise affirms its passage, confirming the presence of one or fewer null types in these columns.

Furthermore, the "Mixed Data Types" check passed, with 21 columns displaying either a significant prevalence of rare data types (exceeding 10%) or a negligible mix of types (below 1%). In particular, none of the 21 columns exhibited a negligible mix of types. The "String Mismatch" check passed for one pertinent column, showing a lack of variations in its string content. The "Data Duplicates" check reported a 0% duplicate data ratio and showed the absence of duplicate records in the dataset. Notably, no relevant columns show string-length outliers, as confirmed by the "String Length Out of Bounds" check.

The "Feature Label Correlation" check passed for 19 relevant columns, signifying that the features' PPS falls below the threshold of 0.8. In the case of "Feature-Feature Correlation", all feature correlations stand below 0.9, with exceptions for a few unmentioned pairs.

However, the "Conflicting Labels" check does not meet the desired criteria, with 0.02% of the samples exhibiting conflicting labels, slightly exceeding the specified threshold of 0%. The results comprehensively evaluate the dataset's quality, highlighting the passage of 9 out of 10 tests and a lone test that falls short by a fractional margin.

Table 6.14: Data purity (Telemetry for Predictive Maintenance)

Status	Check	Condition	More Information
Passed	Single Value in Column	It does not contain only a single value	Passed for 21 relevant columns
Passed	Special Characters	The ratio of samples containing special characters is less or equal to 0.1%	Passed for 21 relevant columns
Passed	Mixed Nulls	The number of different null types is less or equal to 1	Passed for 21 relevant columns
Passed	Mixed Data Types	Rare data types in column are either more than 10% or less than 1% of the data	21 columns passed: found 0 columns with negligible types mix, and 21 columns without any types mix
Passed	String Mismatch	No string variants	Passed for one relevant column
Passed	Data Duplicates	Duplicate data ratio is less or equal to 5%	Found 0% duplicate data
Passed	String Length Out of Bounds	The ratio of string length outliers is less or equal to 0%	No relevant columns to check were found
Passed	Feature-Label Correlation	Features' Predictive Power Score is less than 0.8	Passed for 19 relevant columns

Passed	Feature-Feature Correlation	Not more than 0 pairs are correlated above 0.9	All correlations are less than 0.9 except for pairs
Failed	Conflicting Labels	Ambiguous sample ratio is less or equal to 0%	The ratio of samples with conflicting labels: 0.02%

Table 6.15 affirms column diversity, revealing multiple values. Then, it validates the absence of excessive special characters, ensuring that samples with only these characters remain below 0.1%. Furthermore, it ensures uniform null-value handling, achieving the same results for all pertinent columns. Uncommon data types undergo analysis, which succeeds in six columns, showing minimal data type variations.

In addition, the evaluation inspects string variations, succeeding for a single pertinent column, indicating string data uniformity. Duplicate data undergoes analysis, with no duplications detected. String length outliers lack relevance because no relevant columns were found for this check. The evaluation checks for conflicting labels that pass the test. Feature-feature correlations were analysed, requiring no more than 0 pairs to exhibit correlations exceeding 0.9, which is met. Finally, the evaluation of feature predictive power, identifying three out of four features with robust predictive capabilities, notably “sensor_05”, “sensor_10_window_3H_mean”, and “sensor_12_window_3H_mean”, with scores of 0.93, 0.83, 0.81, respectively. Like the previous dataset, the evaluations showcase the successful passage of 9 out of 10 tests and a single test that narrowly falls short.

Table 6.15: Data purity (Pump Sensor)

Status	Check	Condition	More Information
Passed	Single Value in Column	It does not contain only a single value	Passed for six relevant columns
Passed	Special Characters	The ratio of samples containing special characters is less or equal to 0.1%	Passed for six relevant columns
Passed	Mixed Nulls	The number of different null types is less or equal to 1	Passed for six relevant columns

Passed	Mixed Data Types	Rare data types in column are either more than 10% or less than 1% of the data	Six columns passed: found 0 columns with negligible types mix, and six columns without any types mix
Passed	String Mismatch	No string variants	Passed for one relevant column
Passed	Data Duplicates	Duplicate data ratio is less or equal to 5%	Found 0% duplicate data
Passed	String Length Out of Bounds	The ratio of string length outliers is less or equal to 0%	No relevant columns to check were found
Passed	Conflicting Labels	Ambiguous sample ratio is less or equal to 0%	The ratio of samples with conflicting labels: 0%
Passed	Feature-Feature Correlation	Not more than 0 pairs are correlated above 0.9	All correlations are less than 0.9 except for pairs
Failed	Feature-Label Correlation	Features' Predictive Power Score is less than 0.8	Found 3 out of 4 features with PPS above threshold: sensor_05: 0.93 sensor_10_window_3H_mean: 0.83 sensor_12_window_3H_mean: 0.81

ii. Insights from the model

As presented and discussed in Chapter 2, within the domain of "Approaches to explainability", a focus on global explainability emerged, and further probed into model-centric methodologies that explore links between input features and outcomes; this approach yields rich insights directly from the model, primarily through SHAP plots, which clarify feature impact direction using colour-coded bars (blue for negative, red for positive). For clarity, plot titles like "class 'compY'" indicate "compX's" influence on the target "compY", where X denotes a feature and Y denotes a target variable. An "error count" feature also exists, representing the number of times a specific error type has occurred. For example, "error1count_2" signifies that an error of type 1 has occurred on 2 occasions. Also, the features "datetime_hour_sin" and "datetime_hour_cos" are engineered out of sine and cosine transformations, as thoroughly explained in Chapter 5 under "Periodic Cyclical Features". Unlike feature importance, SHAP values provide a nuanced view of feature attribution, applying game theory for a comprehensive impact assessment (Lundberg & Lee, 2017; Shapley, 1952). Also, unlike feature importance, which ranks features by performance but lacks detailed impact

analysis (Molnar, 2022), SHAP values offer a balanced view, suitable for both global and local effects. The research also leverages SHAP for global feature attribution using DL and ML algorithms in the model dimension, providing a holistic understanding of model behaviour.

a. Deep Learning (Model results)

Subsequently, an analysis ensues on results derived from the model dimension using SHAP on DL algorithms for global feature attribution; this approach adopts a comprehensive perspective on explainability, extending beyond singular prediction instances. Crucially, global feature attribution via SHAP assigns mean SHAP values to features, contrasting with local feature attribution that assigns concrete SHAP values.

Figure 6.1 sheds light on features swaying prediction for class “comp1", signalling machine failure incurred due to comp1. Dominating this prediction, the "error1count_1" feature stood tall with a SHAP value of 0.09. Equally influential for the model's output regarding class “comp1", "datetime_week", and "comp4", each carries a SHAP value of 0.02. On the contrary, "comp3", "age", and "comp2" bear a minor sway with SHAP values resting at 0.01. Numerous features, including the feature "comp1", exhibit zero effect toward the prediction of class “comp1" prediction, as their SHAP value stands at 0.

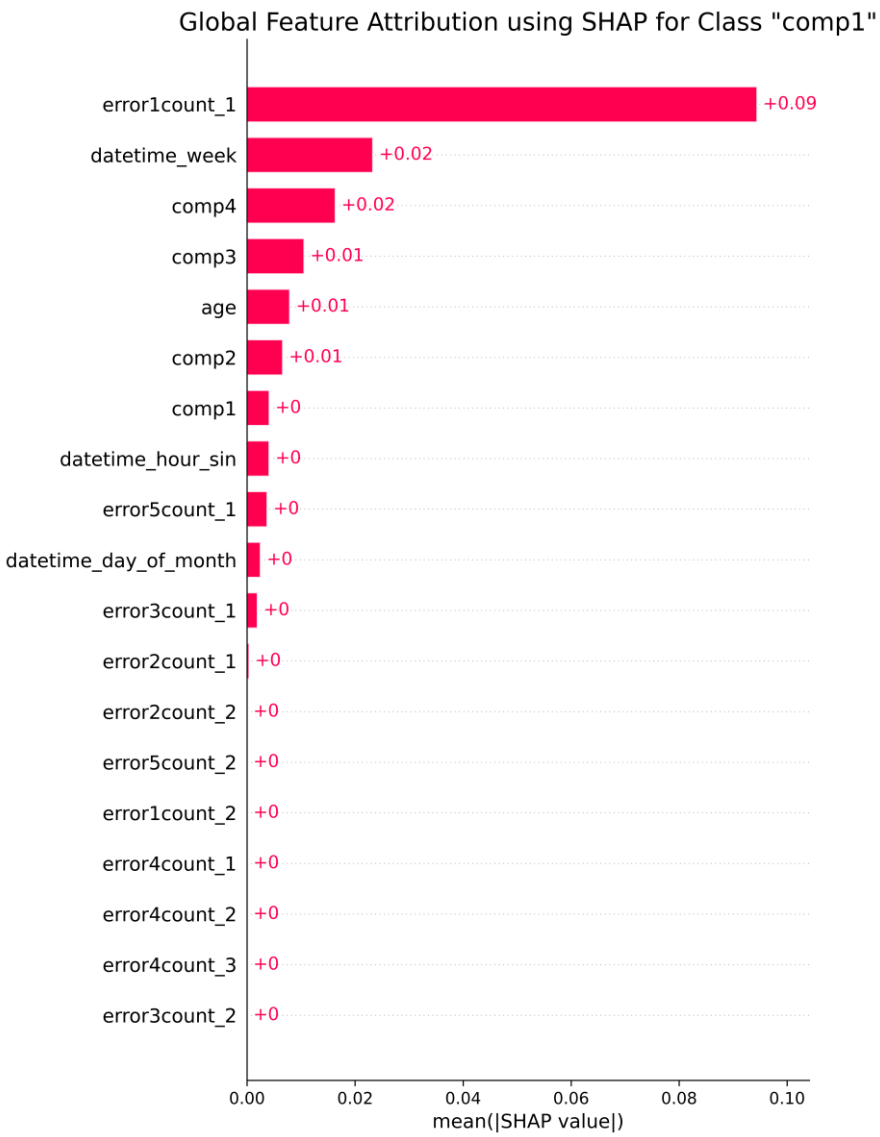

Figure 6.1: DL - Global SHAP for class comp1 (Telemetry for Predictive Maintenance)

The application of SHAP values lies in discerning the impact of various sensor readings on predicting the status of a water pump, specifically its "broken" state. In Figure 6.2, “sensor_05” boasts the highest SHAP value of 0.02, exerting the most effect toward this prediction, thereby showing its paramount role in determining a "broken" pump status. Following closely, “sensor_10_window_3H_mean”, representing an average reading from “sensor_10” over a three-hour window, holds a SHAP value of 0.01, indicating a noteworthy but lesser contribution to the prediction than “sensor_05”. Features “sensor_11_window_3H_mean” and “sensor_12_window_3H_mean”, with SHAP values of 0, hint at their negligible effect towards predicting a "broken" pump status. Consequently, this model

identifies “sensor_05” and “sensor_10_window_3H_mean” as crucial features in predicting a "broken" pump status; this insight could steer future data collection or feature engineering efforts, possibly with an increased focus on sensors 05 and 10.

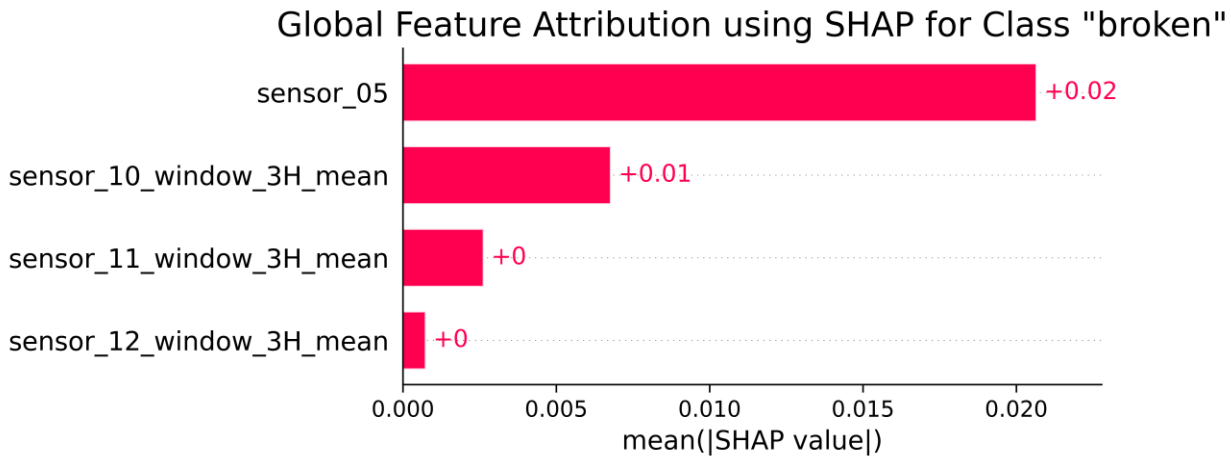

Figure 6.2: DL - Global SHAP for class broken (Pump Sensor)

b. Machine Learning (Model results)

Following on, an analysis unfolds on the model dimension obtained from employing SHAP on ML algorithms for global feature attribution. As seen in Figure 6.3, “error1count_1” and “comp1” emerged as features with the highest SHAP values, precisely 0.01; this underscores their paramount significance in predicting class “comp1". These features, thus, hold substantial sway over the model's output. Consequently, alterations in these feature values considerably affect the predicted outcome for "comp1".

On the contrary, the remaining features in the model registered a SHAP value of 0, indicating their negligible contribution towards class "comp1" prediction. In essence, variations in these feature values have a minimal effect on the predicted outcome for class "comp1".

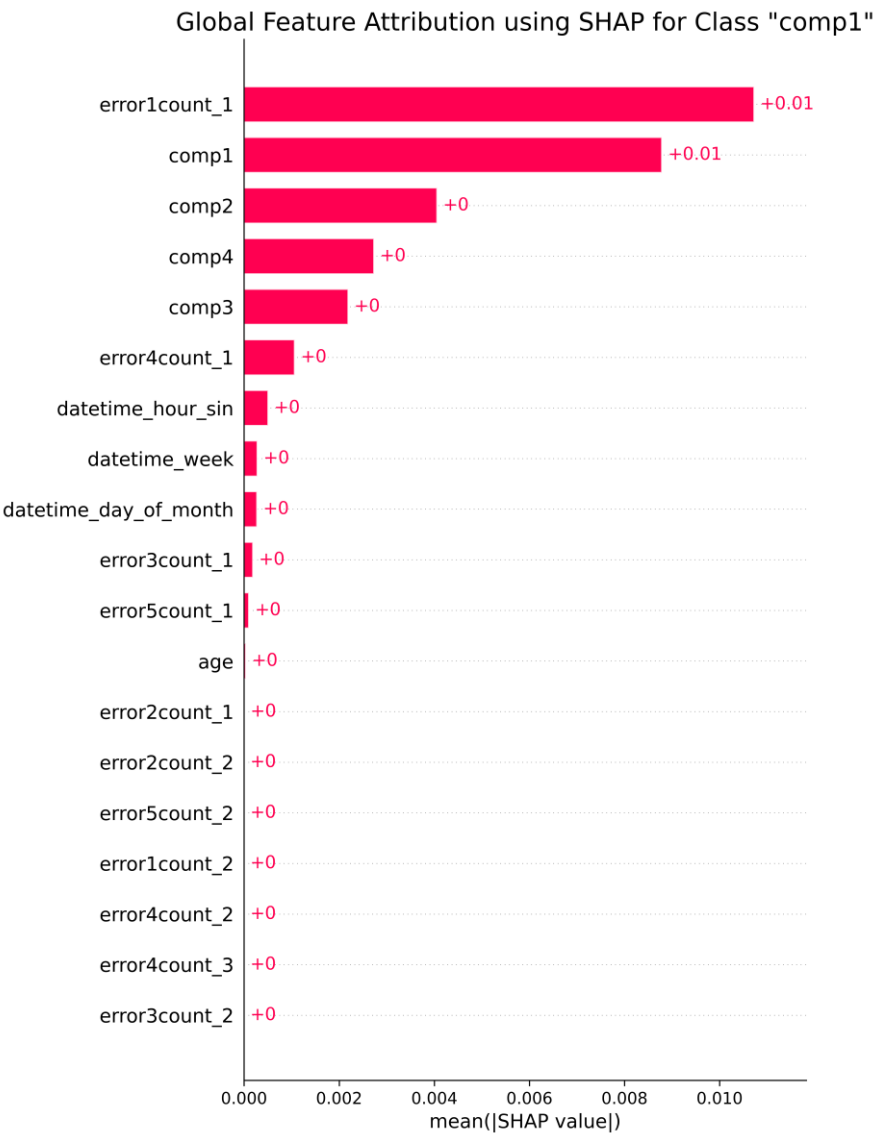

Figure 6.3: ML - Global SHAP for class comp1 (Telemetry for Predictive Maintenance)

With an impactful SHAP value of 0.04, "sensor_05" is a crucial feature; its higher readings boost the effect toward pump status classification as "broken" in Figure 6.4. The features "sensor_11_window_3H_mean" and "sensor_10_window_3H_mean", both displaying SHAP values of 0.01, contribute positively to the prediction, although with less potency than "sensor_05".

On the contrary, "sensor_12_window_3H_mean", bearing a SHAP value of 0, exerts no average impact on the prediction, yet retains the potential for affecting individual predictions. While all features, except for "sensor_12_window_3H_mean", augment the effect towards pump status classification as "broken", impacts register as relatively weak, given their small SHAP values; this

observation underscores the impact of not overlooking the potential effect of features with low SHAP values.

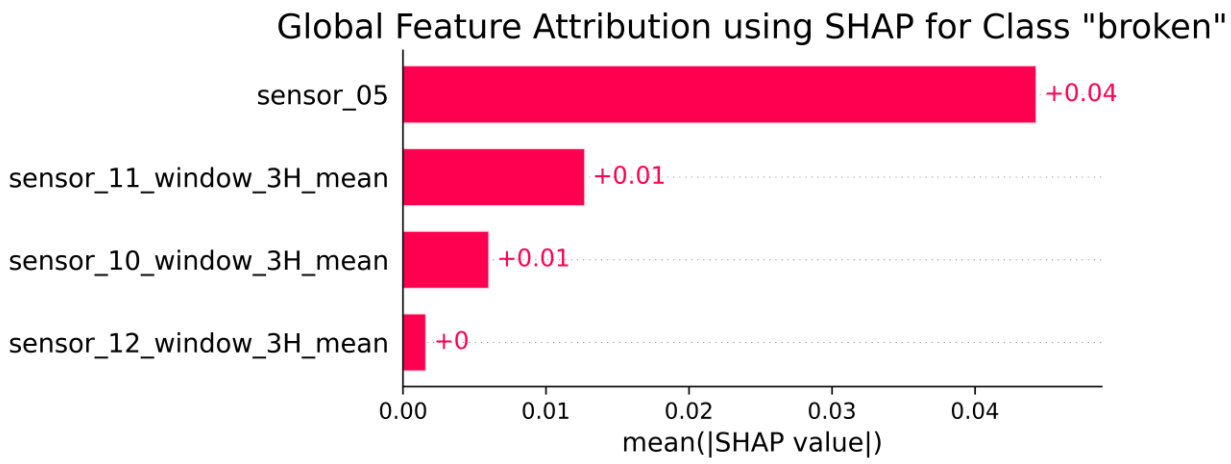

Figure 6.4: ML - Global SHAP for class broken (Pump Sensor)

iii. Insights from the outcome

In Chapter 2, explaining "Approaches to explainability", the focus shifted to local explainability. Here, local explainability methods illuminate individual prediction instances, extending the model's insights. SHAP, central to this analysis, studies single-instance feature impact. Features like "error1count_2" quantify specific error occurrences, while "datetime_hour_sin" and "datetime_hour_cos" emerge from sine and cosine transformations (detailed in Chapter 5 's "Periodic Cyclical Features"). The research then applied SHAP to DL and ML for local feature attribution in the outcome dimension, providing detailed insights. Unlike global attribution, local SHAP provides granular explainability.

a. Deep Learning (Outcome results)

Next, analysis of the outcome dimension through SHAP's application on DL algorithms for local feature attribution; this approach embraces isolated prediction instances, permitting granular level explainability. In particular, local feature attribution via SHAP assigns specific SHAP values to features, avoiding mean SHAP value allocation. Note that the left-hand side of each feature reveals its actual value for that prediction instance, providing enhanced detail and granularity.

In the instance of Figure 6.5, one observes "error1count_1" with an instance value of 1 exerting a substantial positive effect, augmenting the propensity for classification as "comp1" by 0.59. "Comp4", possessing an instance value of 4.582, also strengthens this propensity, although with diminished intensity, reflected by a SHAP value of 0.08. "Comp2", with an instance value of 0.294, exerts a minor positive effect, denoted by a SHAP value of 0.02.

On the contrary, "datetime_week" (instance value: 1.686), "datetime_hour_sin" (instance value: -0.001), and "comp3" (instance value: -0.429), all diminish the effect toward the classification as "comp1", with respective SHAP values of -0.02, and -0.01, -0.01.

The "age" feature, with an instance value of 0.973, exerts a negligible positive effect on classification, as indicated by a SHAP value of 0.01. Last, the features "error3count_1", "comp1", "error5count_1", "error2count 2", "error2count_1", "error5count_2", "error1count_2", "error4count_1", "error4count_2", "error4count_3", and "error3count_2" exert no effect on the classification as “comp1”, evidenced by their SHAP values of 0.

In essence, the magnitude of the SHAP value signifies the strength of the effect of the feature on the prediction outcome, with positive values increasing the propensity of the class, negative values diminishing it, and values close to 0 indicating minimal or no impact.

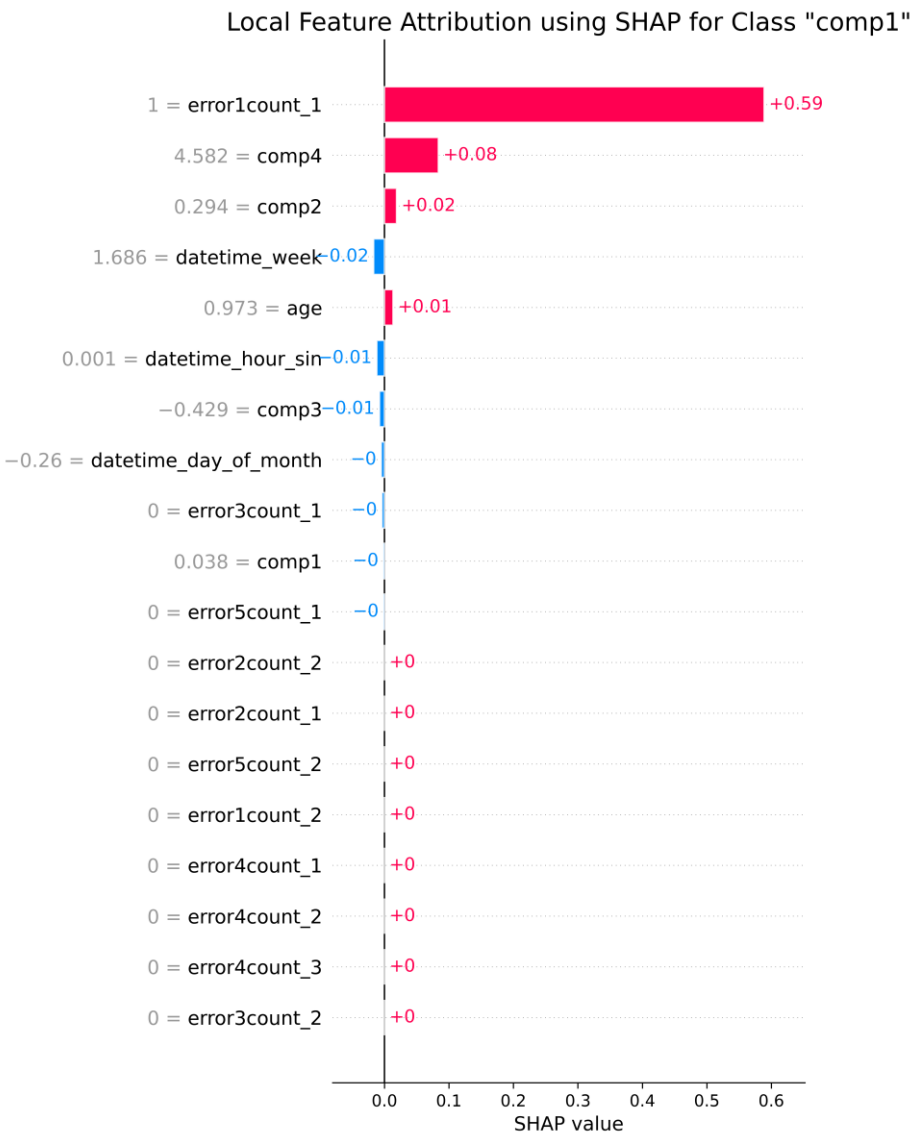

Figure 6.5: DL - Local SHAP for class comp1 (Telemetry for Predictive Maintenance)

As depicted in Figure 6.6, feature "sensor_05" possesses a SHAP value of -0.02, representing its instance value of -0.106, which diminishes the classification effect towards the "broken" pump status, evidenced by the negative SHAP value. Conversely, "sensor_10_window_3H_mean" with SHAP value 0.01 indicated that an instance value of 0.411 escalates the classification effect towards the "broken" status. Notably, the features "sensor_11_window_3H_mean" and "sensor_12_window_3H_mean" exhibit SHAP values of 0, signifying that their given values exert no effect on predictions for the "broken" state.

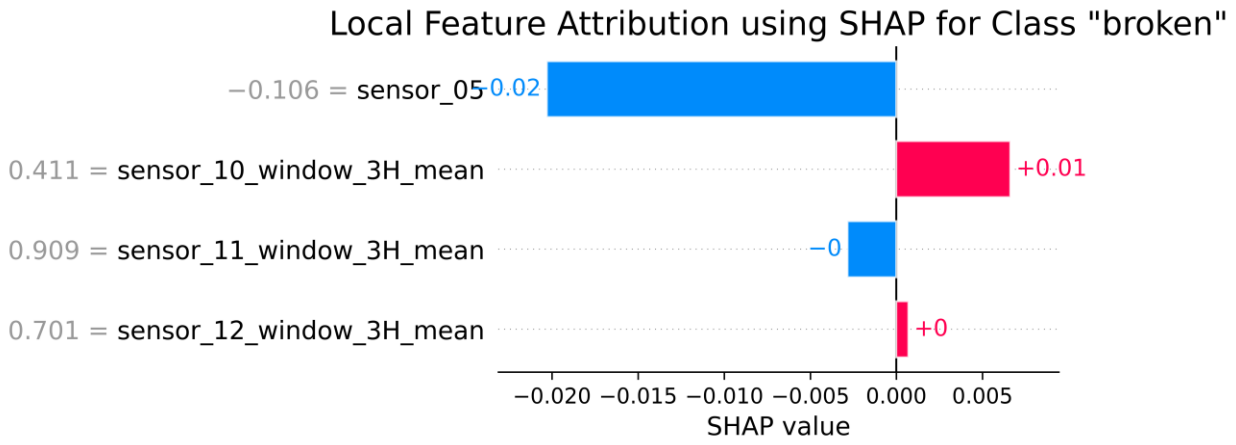

Figure 6.6: DL - Local SHAP for class broken (Pump Sensor)

b. Machine Learning (Outcome results)

In the following analysis, the focus shifts to the outcome dimension arising from applying SHAP to ML algorithms for local feature attribution. In Figure 6.7, “error1count_1” emerged as an influential feature, with an instance value of 0 that impacts the prediction towards class "comp1" by a SHAP value of 0.41. The feature "comp1" followed closely, impacting the class "comp1" prediction by a SHAP value of 0.24 with an instance value of 0.038. Both "comp3" and "comp4" contribute similarly to "comp1" class prediction, augmenting it by a SHAP value of 0.1 with instance values of -0.429 and 4.582, respectively. "Comp2" with an instance value of 0.294 boosts the class "comp1" prediction by a SHAP value of 0.09. "Datetime_hour_sin" with an instance value of 0.001 impacts the class "comp1" prediction by a SHAP value of 0.04. Finally, both "error4count_1" and "error3count_1" impact the prediction of the class “comp1” by a SHAP value of 0.01 with an instance value of 0. The remaining features did not affect the class "comp1" prediction for given instance values.

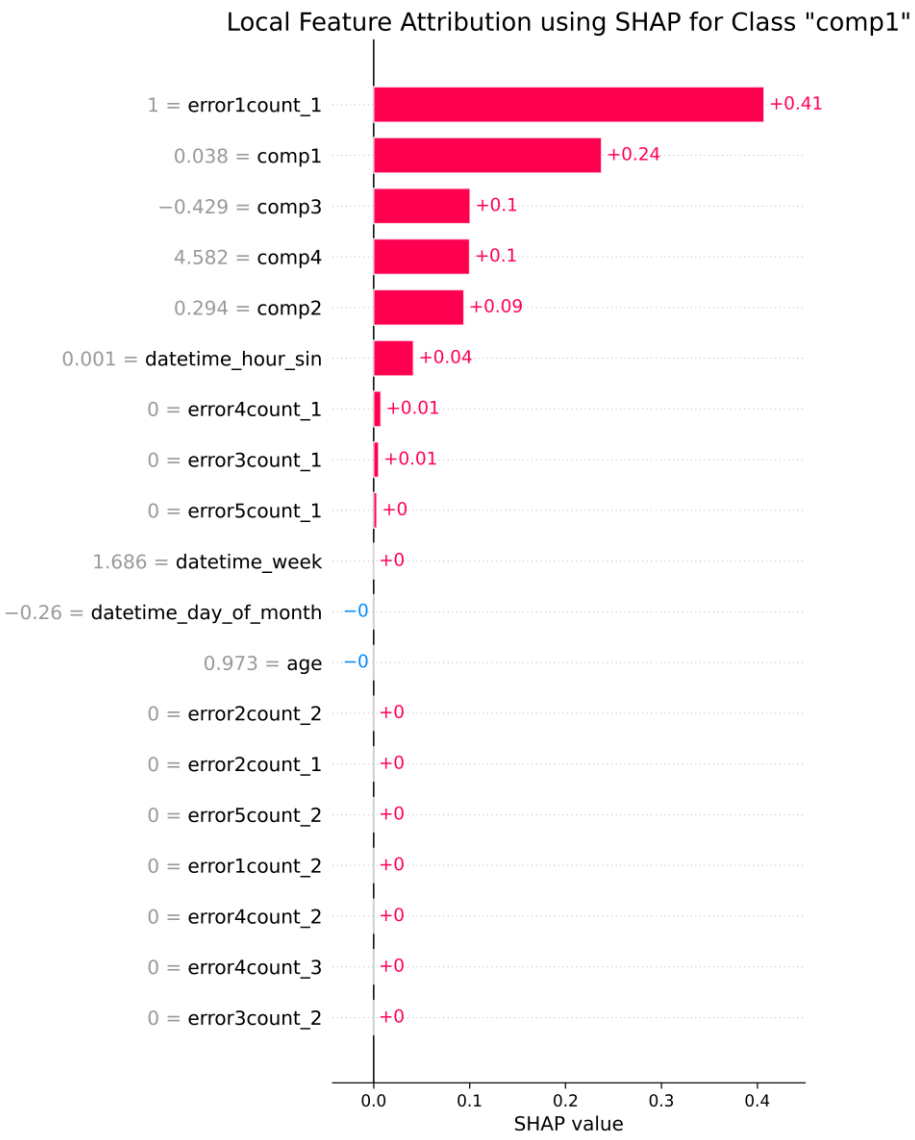

Figure 6.7: ML - Local SHAP for class comp1 (Telemetry for Predictive Maintenance)

In Figure 6.8, the feature "sensor_11_window_3H_mean" with an instance value of 0.909 exhibited a positive SHAP value of 0.01; this signifies an elevation in the feature's value augments the potential for the pump's malfunction (prediction class “broken”). Similarly, "sensor_12_window_3H_mean" with an instance value of 0.701 also manifests a positive SHAP value, indicating that an escalation in this feature's value enhances the potential for pump failure.

Conversely, "sensor_05", with an instance value of -0.106, demonstrated a negative SHAP value of -0.01, indicating that reducing this feature's value heightens the potential for pump failure.

Finally, "sensor_10_window_3H_mean" with an instance value of 0.411 shows a positive SHAP value, denoting an increase in this feature's value would boost the potential for pump failure.

In essence, SHAP values offer insights into each feature's effect on the prediction from the base value. A positive SHAP value escalates the potential for the class to be "broken", while a negative value diminishes it. The magnitude of the SHAP value embodies the strength of that feature's effect. In this case, most features manifest the same effect magnitude, but the effect direction (positive or negative) changes with the feature; this information proves vital in discerning which features hold a paramount effect in predicting a water pump's malfunction.

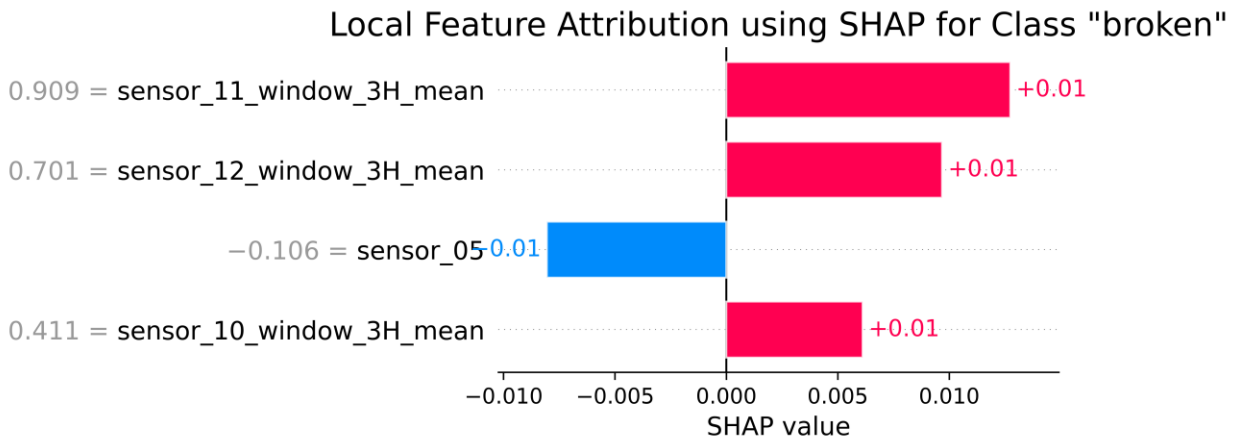

Figure 6.8: ML - Local SHAP for class broken (Pump Sensor)

iv. Insights toward the end-user

In Chapter 2, exploring the "Dimensions of explainability", the imperative task emerged: to create explanations that strike the proper equilibrium between abstraction and detail for the target audience. Therefore, the ensuing section reveals the results of applying counterfactual approaches meticulously tailored to yield user-centric explanations. "Counterfactual n for class 'compY'" indicates altering "compX" to trigger class "compY", where X denotes a feature, and Y denotes a target variable. Phrases like "The feature compX must transition" indicate necessary changes in "compX" to affect class "compY". Some CFEs, without percentage change, relate to categorical features, while others, exceeding 100%, indicate the extent of change needed for a desired outcome. Light grey and light blue in the CFE tables represent decreasing and increasing feature changes, respectively.

a. Deep Learning (End-user results)

Table 6.16 outlined a range of “what-if” scenarios to change prediction class “none” to class “comp1" for the Telemetry for Predictive Maintenance dataset. Each scenario depicts changes in specific features and their corresponding percentage changes. In particular, Counterfactual one and Counterfactual four demonstrated that adjustments to features such as "comp4" and "comp2" result in substantial percentage variations (an increase of 593.70% and a decrease of -165.64%, respectively) in the outcome; this showed that these features exert a crucial effect on class “comp1”. More so, in Counterfactuals one through four, the transition of the "errorXcount_Y" feature from 0.0 to 1.0 underscored the impact of the presence or absence of specific errors on the final result. Although Counterfactual three showed a relatively modest percentage change (22.33%) in the "datetime_week" feature, it underscores the potential significance of even minor adjustments.

Furthermore, Counterfactual five underscored the consistent effect of "error1count_2" and "error2count_1" on the outcome, as they remain consistently set to 1.0. The interaction between these features and their cumulative effect on the target minimises feature values, shifting predictions toward a desired outcome, and providing an assortment of choices for end-users to choose from.

Table 6.16 DL - CFE from none to comp1 (Telemetry for Predictive Maintenance)

Counterfactual one for class “comp1"
The feature comp4 must transition from -0.369327187538147 to 1.82338613 **(593.70%)**
The feature error4count_2 must transition from 0.0 to 1.0
Counterfactual two for class “comp1"
The feature comp1 must transition from 2.1010522842407227 to 4.61362886 **(119.59%)**
The feature error5count_2 must transition from 0.0 to 1.0
Counterfactual three for class “comp1"
The feature datetime_week must transition from -1.7300148010253906 to -1.3437247 **(22.33%)**
The feature error5count_2 must transition from 0.0 to 1.0
Counterfactual four for class “comp1"
The feature comp2 must transition from 1.2788968086242676 to -0.83951104 **(-165.64%)**

The feature error1count_1 must transition from 0.0 to 1.0
Counterfactual five for class “comp1"
The feature error1count_2 must transition from 0.0 to 1.0
The feature error2count_1 must transition from 0.0 to 1.0

In the Pump Sensor dataset, a shift occurs from the "normal" class to the "broken" class in the predictions. Table 6.17 revealed counterfactuals exposing specific feature alterations. Counterfactual one showcased a substantial feature change, as “sensor_05” must transition from 0.195 to -3.559, indicating a notable decrease of -1924.80%. Counterfactuals two and three introduced adjustments to “sensor_12_window_3H_mean” and “sensor_05”, resulting in similarly noteworthy percentage reductions. Counterfactuals 4 and 5 exclusively modify “sensor_05”, with equally notable percentage declines. Notably, “sensor_05” emerged as a pivotal feature, consistently altered in all scenarios, emphasising its substantial impact on classifying the "broken" class.

Table 6.17 DL - CFE from normal to broken (Pump Sensor)

Counterfactual one for class “broken"
The feature sensor_05 must transition from 0.1950564682483673 to -3.5593897 **(-1924.80%)**
Counterfactual two for class “broken"
The feature sensor_12_window_3H_mean must transition from 0.7513946890830994 to -2.0743027 **(-376.06%)**
The feature sensor_05 must transition from 0.1950564682483673 to -3.0883916 **(-1683.33%)**
Counterfactual three for class “broken"
The feature sensor_12_window_3H_mean must transition from 0.7513946890830994 to -2.2550662 **(-400.12%)**
The feature sensor_05 must transition from 0.1950564682483673 to -3.3266737 **(-1805.49%)**
Counterfactual four for class “broken"
The feature sensor_05 must transition from 0.1950564682483673 to -3.639838 **(-1966.04%)**
Counterfactual five for class “broken"

The feature sensor_05 must transition from 0.1950564682483673 to -3.523411 **(-1906.35%)**

b. Machine Learning (End-user results)

Table 6.18 shows counterfactuals to change class "none" to class "comp1". In the initial counterfactual, a significant reduction in "datetime_day_of_month" and "comp4" features, coupled with the inclusion of "error1count_2", brought about a notable shift, underscoring the substantial effect of these features on the class. In the second counterfactual, a substantial increase in "comp1" and the incorporation of error counts ("error1count_1" and "error2count_1") yielded a profound transformation in the class "comp1", emphasising their criticality. Counterfactual three showed that a moderate increase in the feature "comp1" and the introduction of error counts ("error1count_1" and "error2count_2") resulted in a noticeable change. Counterfactuals four and five revealed the model's increased sensitivity to specific features, particularly for the class "comp1".

Notably, Counterfactuals one and two showcased the substantial effect of features such as "comp1", "datetime_day_of_month", and "comp4" on the model's predictions, evidenced by substantial percentage shifts. Furthermore, introducing error counts, such as "error1count_1" and "error2count_1", from 0.0 to 1.0, was associated with pronounced changes in class predictions, underscoring the importance of error-related variables.

Table 6.18 ML - CFE from none to comp1 (Telemetry for Predictive Maintenance)

Counterfactual one for class "comp1"
The feature datetime_day_of_month must transition from -0.2595066428184509 to -1.5150433 **(-483.82%)**
The feature comp4 must transition from -0.2830199897289276 to -0.73243351 **(-158.79%)**
The feature error1count_2 must transition from 0.0 to 1.0
Counterfactual two for class "comp1"
The feature comp1 must transition from -0.47939661145210266 to 3.43122288 **(815.74%)**
The feature error1count_1 must transition from 0.0 to 1.0
The feature error2count_1 must transition from 0.0 to 1.0
Counterfactual three for class "comp1"

The feature comp1 must transition from -0.47939661145210266 to 1.73483364 **(461.88%)**
The feature error1count_1 must transition from 0.0 to 1.0
The feature error2count_2 must transition from 0.0 to 1.0
Counterfactual four for class “comp1"
The feature datetime_day_of_month must transition from -0.2595066428184509 to -0.977336 **(-276.61%)**
The feature error1count_2 must transition from 0.0 to 1.0
Counterfactual five for class “comp1"
The feature comp1 must transition from -0.47939661145210266 to 0.13429292 **(128.01%)**
The feature comp3 must transition from -0.505618691444397 to -0.0257899 **(94.90%)**
The feature error1count_1 must transition from 0.0 to 1.0

Table 6.19 showed altering a prediction from "normal" to "recovering" in the Pump Sensor dataset. Counterfactual one demonstrated that changing "sensor_05" from 0.0007093246676959097 to -3.665019 is essential to classify an instance as "recovering”, highlighting this feature's pivotal role. Counterfactual two, on the contrary, unveils the combined impact of "sensor_12_window_3H_mean" and "sensor_05”, as these features necessitate significant adjustments to achieve the same classification. Counterfactuals 3 and 4 also underscore the significance of "sensor_05" and "sensor_10_window_3H_mean" in determining the class. Lastly, Counterfactual five reaffirms the effect of "sensor_05" and "sensor_12_window_3H_mean" in classifying instances as "recovering”. These discoveries signify that specific features, especially "sensor_05”, exert a substantial effect on classification from "normal" to "recovering", and even minor modifications to these features can yield noteworthy changes in the classification outcome.

Table 6.19 ML - CFE from normal to recovering (Pump Sensor)

Counterfactual one for class “recovering"
The feature sensor_05 must transition from 0.0007093246676959097 to -3.665019 **(-516791.32%)**
Counterfactual two for class “recovering"
The feature sensor_12_window_3H_mean must transition from 0.524188756942749 to

-2.5082043 **(-578.49%)**
The feature sensor_05 must transition from 0.0007093246676959097 to -3.657784 **(-515771.34%)**
Counterfactual three for class “recovering"
The feature sensor_05 must transition from 0.0007093246676959097 to -3.5593897 **(-501899.79%)**
The feature sensor_10_window_3H_mean must transition from 0.6247089505195618 to -2.8465657 **(-555.66%)**
Counterfactual four for class “recovering"
The feature sensor_05 must transition from 0.0007093246676959097 to -3.6582662 **(-515839.32%)**
The feature sensor_10_window_3H_mean must transition from 0.6247089505195618 to -2.9265728 **(-568.47%)**
Counterfactual five for class “recovering"
The feature sensor_12_window_3H_mean must transition from 0.524188756942749 to -2.4914598 **(-575.30%)**
The feature sensor_05 must transition from 0.0007093246676959097 to -3.6344541 **(-512482.31%)**

6.4 Discussion

After presenting the experimental results, the results are then discussed in the context of this research. It starts with a comparative analysis with related studies while discussing the findings in a similar order as was done in the presentation of results: PdM and the explainable ML results. The first (PdM) discusses the results of both the DL and ML models. The second (explainable ML) discusses the results of explainable ML, focusing on the four dimensions of explainability extracted from the literature: (1) data, (2) model, (3) outcome, and (4) end-user.

6.4.1 Comparative analysis with related studies

The comparative analysis in this section offers insight into the unique contributions of this research, contrasting with previous studies to deepen understanding. Performance comparisons in Table 6.20

demonstrate the superior functioning of DL and ML classifiers on two datasets. The Convolutional LSTM and BiRNN LSTM Neural Nets, excel across metrics without overfitting, highlighting their reliable generalisation. Comparisons with (Wu et al., 2021) show that LSTM classifiers from this research consistently outperformed others, with a test accuracy of 5.81% higher, calculated as an average percentage difference with related studies. Moreover, XGBoost classifiers perform exceptionally well on diverse datasets (Table 6.21). For instance, in the Telemetry for Predictive Maintenance dataset, XGBoost achieves consistent F1 scores and accuracy. Compared to (Steurtewagen & Van den Poel, 2021), this research's XGBoost classifier shows enhanced performance, with an increase of 7.09% in test F1 score, 10.66% in accuracy, and 4.29% in ROC-AUC; this research's XGBoost classifier surpasses (Steurtewagen & Van den Poel, 2021) in all test metrics, affirming the superiority of this approach in PdM, machine status, and water pump status prediction. These results emphasise the efficacy of LSTMs and Boosters for PdM in SAF.

Table 6.20: DL - classifier comparison with related studies

Source	Dataset	Classifier	Train F1 score	Test F1 score	Overfitting	Train accuracy	Test accuracy	Test ROC-AUC
This research	Telemetry for Predictive Maintenance	Convolutional LSTM Neural Net	0.958015	0.970003	False	0.942831	0.942831	**0.967616**
This research	Telemetry for Predictive Maintenance	BiRNN LSTM Neural Net	0.95516	0.963522	False	0.945424	0.945424	0.957594
This research	Pump Sensor	BiRNN LSTM Neural Net	0.862548	**0.998576**	False	0.90576	**0.998782**	0.928758
This research	Pump Sensor	Convolutional LSTM Neural Net	0.955991	0.960525	False	0.933075	0.92518	0.935467
(Wu *et al.*, 2021)	Time series	RNN LSTM	-	-	-	-	0.9007	-

Table 6.21: ML - classifier comparison with related studies

Source	Dataset	Classifier	Train F1 score	Test F1 score	Overfitting	Train accuracy	Test accuracy	Test ROC-AUC
This research	Telemetry for Predictive Maintenance	XGB classifier	0.993043	0.993304	False	0.993733	0.993733	**0.993779**
This research	Pump Sensor	XGB classifier	0.995269	**0.998347**	False	0.995255	**0.998051**	0.906245
(Steurtewagen & Van den Poel, 2021)	Time series	XGB classifier	-	0.93	-	-	0.9	0.911

i. Data dimension

Contrary to related studies, this research addresses an unexplored dimension: data explainability. Pioneering in this dimension, this study sets a precedent for future research. It demonstrates how to discern data limitations and align researcher expectations. Tables 6.14 and 6.15 exhibit a rigorous analysis of the dataset, highlighting diversity, uniformity, and minimal duplications. Despite minor label discrepancies, these datasets (Telemetry for Predictive Maintenance and Pump Sensor) exemplify consistency and integrity, crucial for reliable PdM and ML analysis.

ii. Model and Outcome dimensions

This research diverges from related studies in its model and outcome explainability approach. A key premise often assumed is the independence and non-correlation of features or attributes; this assumption is fallible, introducing potential biases in the "model and outcome dimensions". Thus, data explainability becomes a foundational element for these dimensions, guiding the assessment of inherent limitations and depth of insights derivable from the data; this research's contribution lies in establishing a framework for exploring the levels of explainability achievable with specific datasets. Employing SHAP for global feature attribution in DL and ML, it transcended mere feature ranking, enabling a comprehensive understanding of model behaviour. In local

explainability, the research goes further, analysing individual prediction instances. Local methods, through SHAP, provide an in-depth feature impact analysis on single-instance predictions.

iii. End-user dimension

Concerning end-user explainability, this research stands out from existing literature, particularly in explaining predictions to non-technical users; this research significantly advances end-user explainability, showcasing how to strike this balance effectively. Using the two datasets, it presented "what-if" (counterfactuals) scenarios in Tables 6.16, 6.17, 6.18 and 6.19, demonstrating the necessary changes to shift predictions from one class to another preferred class (outcome). These counterfactuals elucidate the impact of specific features, underscoring the significance of consistent settings in influencing outcomes.

6.5 Summary

This chapter dissected data, exposing insights into PdM and elucidating the intricacies of explainable ML. Methodically segmented into specific sub-themes, the analysis considered prognostics and diagnostics, adeptly predicting system and component failures. A particular highlight was the precision in predicting the water pump's condition and identifying the component responsible for machine failure. The prowess of the DL and ML algorithms was evident in both prognostics and diagnostics. Exploring "data explainability", a data-centric approach navigated challenges and expectations inherent in extracting insights. “Model and outcome explainability” strategies took centre stage, unravelling the model's behaviour and offering insights into global and singular prediction instances. In addition, this chapter underscored the imperative of a balanced approach to creating explanations for the "end-user dimension". The forthcoming chapter encapsulates the conclusion of this comprehensive research dissertation.

CHAPTER 7 CONCLUSION

7.1 Overview

Chapter 1's problem statement highlighted the lack of transparency and explainability in ML models in SAF, leaving farmers grappling with indefinite implications; this research advocated for an amalgamated model, fusing explainable ML with PdM, thereby augmenting understanding and applicability within SAF. Therefore, it aimed to (1) predict maintenance needs and (2) provide explanations using explainable ML for PdM in SAF. Consequently, this chapter presents responses to research objectives and questions aligned with the research aim, key contributions, and future research directions and recommendations.

7.2 Response to research objectives and questions

This research contributes notable insights into explainable ML for PdM in SAF by addressing the research questions and fulfilling the established objectives. These comprise the following.

i. What is the current landscape of ML models and the need for explainable Machine Learning?

Chapter 2's bibliometric analysis laid bare a landscape dominated by traditional ML models in PdM, corroborating findings from related literature. However, this is just the tip of the iceberg. What is genuinely compelling here, and underemphasised, is the emerging integration of explainable ML in the areas of DL; this is not just a passing trend but a call for a paradigm shift. The fusion of explainability into DL's complex, often opaque algorithms is not just desirable but also downright essential. It is about shedding light on the murky waters where DL algorithms swim, ensuring that the decisions these models make are not just accurate, but also transparent and understandable; this is the new frontier: a world where DL does not just perform with precision, but also with clarity and accountability.

ii. How should one develop an ML model to predict maintenance needs?

This research advances the existing body of knowledge by overcoming the shortcomings highlighted in the literature review, as discussed in Chapter 3 and detailed in Chapter 5 in the section "Research design" of stages 2 to 6. Here, the focus changed from pedestrian classification accuracy measures to a more transparent and rigorous analysis. By unveiling both training and testing outcomes, the research

courageously exposes areas where models might stumble, ensuring that there are no skeletons in the closet. The ROC-AUC metric offers a stricter evaluation of DL and ML. The results strongly endorse the use of LSTM for DL and Boosters for ML for PdM tasks. The rationale? Boosters emerged as titans for ML, while LSTMs led the charge for DL algorithms, which are critical for SAF datasets.

iii. How does the proposed ML model perform?

Compared to the related literature, the proposed ML model exhibited superior performance, exceeding benchmarks set by their counterparts, as shown in Chapter 6 in the "Predictive Maintenance results" section. LSTM classifiers demonstrated a noteworthy improvement, with test accuracy exceeding related studies by 5.81%; this result underscores the efficacy of DL models, particularly within PdM for SAF. Furthermore, the XGBoost classifier from this research surpassed related studies, exhibiting superior test metrics; this performance improvement manifested itself as an increase in the test F1 score by 7.09%, accuracy by 10.66%, and ROC-AUC by 4.29%; such results attest to the superiority of the proposed approach and its contribution to PdM, machine status, and water pump status prediction.

iv. What and why does the proposed ML model make specific predictions?

Exploring data dimensions reveals inherent limitations and refines expectations for data-driven insights. The research unravels the model and outcome complexities using SHAP values and providing detailed, practical explanations. It also introduces varied predictive scenarios through counterfactuals, which could improve stakeholder decision-making.

7.3 Key research contributions

This research notably advances ML by conducting a comprehensive SLR, highlighting the critical need for explainable ML beyond traditional accuracy metrics. It pioneers the integration of explainable ML and PdM in SAF, addressing four dimensions of explainability: (1) data, (2) model, (3) outcome, and (4) end-user; this paradigm shift aligns with stakeholder expectations, marking a comprehensive approach in the agricultural sector. The theoretical contribution of the research lies in adapting design knowledge from explainable ML and PdM to the SAF context.

7.4 Future research directions and recommendations

This research detects diverse scopes for future studies and enhancement. The recommended directions for future research foster a coherent flow, starting from practical implementations involving human collaboration, and concluding with a broader assessment of ethical and long-term implications, setting the stage for future studies that are as impactful as they are essential.

- Multi-modal data integration: Synthesise diverse data types, encompassing textual and imaging data, to augment the efficacy of ML models in SAF.

- Explainability metrics: Devise novel metrics to measure the explainability of ML models in SAF, transcending conventional performance metrics such as accuracy, F1 score, and ROC-AUC.

- Human-in-the-Loop systems: Analyse the incorporation of Human-in-the-Loop (HITL) systems in SAF. Construct frameworks where human expertise complements the ML pipeline, facilitating collaborative decision-making and strengthening the overall dependability of PdM models.

- Human-AI interaction in agriculture: Probe the impact of integrating explainable ML into human decision-making processes amid agricultural operations, considering user feedback and interaction patterns.

- Ethical considerations in agricultural ML: Analyse the ethical ramifications of deploying ML models in agriculture, focusing on Fairness, Accountability, and Transparency (FAT) in decision-making algorithms.

- Long-term impact assessment: Assess the long-term impact of deploying explainable ML and PdM in SAF on productivity, sustainability, and economic factors.

BIBLIOGRAPHY

Akoka, J., Comyn-Wattiau, I., Prat, N. & Storey, V.C. 2023. Knowledge contributions in design science research: Paths of knowledge types. *Decision Support Systems*, 166:1-14. 10.1016/j.dss.2022.113898

Aria, M. & Cuccurullo, C. 2017. An R-tool for comprehensive science mapping analysis. *Journal of Informetrics*, 11(4):959-975. 10.1016/j.joi.2017.08.007

Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, 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. 10.1016/j.inffus.2019.12.012

Bell, L.W., Moore, A.D. & Kirkegaard, J.A. 2014. Evolution in crop-livestock integration systems that improve farm productivity and environmental performance in Australia. *European Journal of Agronomy*, 57:10-20. 10.1016/j.eja.2013.04.007

Brauneck, A., Schmalhorst, L., Kazemi Majdabadi, M.M., Bakhtiari, M., Volker, U., Baumbach, J., ... Buchholtz, G. 2023. Federated Machine Learning, Privacy-Enhancing Technologies, and Data Protection Laws in Medical Research: Scoping Review. *J Med Internet Res*, 25:1-18. https://www.ncbi.nlm.nih.gov/pubmed/36995759 10.2196/41588

Breiman, L. 1996. Bagging predictors. *Machine Learning*, 24(2):123-140. 10.1007/bf00058655

Breiman, L. 1999. Pasting small votes for classification in large databases and on-line. *Machine Learning*, 36:85-103. 10.1023/A:1007563306331

Breiman, L. 2001. Random forests. *Machine Learning*, 45:5-32. 10.1023/A:1010933404324

Breiman, L., Friedman, J., Olshen, R.A. & Stone, C.J. 1984. *Classification and Regression Trees*. 1st ed. Chapman and Hall/CRC.

Brinkrolf, J. & Hammer, B. 2018. Interpretable machine learning with reject option. *At-Automatisierungstechnik*, 66(4):283-290. 10.1515/auto-2017-0123

Caelen, O. 2017. A Bayesian interpretation of the confusion matrix. *Annals of Mathematics and Artificial Intelligence*, 81:429-450. 10.1007/s10472-017-9564-8

Calcante, A., Fontanini, L. & Mazzetto, F. 2013. Repair and Maintenance Costs of 4wd Tractors in Northern Italy. *Transactions of the Asabe*, 56(2):355-362. 10.13031/2013.42660

Carrington, A.M., Manuel, D.G., Fieguth, P.W., Ramsay, T., Osmani, V., Wernly, B., ... Holzinger, A. 2023. Deep ROC Analysis and AUC as Balanced Average Accuracy, for Improved Classifier Selection, Audit and Explanation. *IEEE Trans Pattern Anal Mach Intell*, 45(1):329-341. https://doi.org/10.1109/TPAMI.2022.3145392 10.1109/TPAMI.2022.3145392

Chen, T. & Guestrin, C. 2016. *XGBoost: A Scalable Tree Boosting System*. Paper presented at the Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA. https://doi.org/10.1145/2939672.2939785 Date of access: 7 September 2023.

Chhetri, T.R., Kurteva, A., DeLong, R.J., Hilscher, R., Korte, K. & Fensel, A. 2022. Data Protection by Design Tool for Automated GDPR Compliance Verification Based on Semantically Modeled Informed Consent. *Sensors*, 22(7):1-35. https://doi.org/10.3390/s22072763 10.3390/s22072763

Choi, H., Kim, D., Kim, J., Kim, J. & Kang, P. 2022. Explainable anomaly detection framework for predictive maintenance in manufacturing systems. *Applied Soft Computing*, 125:1-17. 109147. 10.1016/j.asoc.2022.109147

Chorev, S., Tannor, P., Israel, D.B., Bressler, N., Gabbay, I., Hutnik, N., ... Rokach, L. 2022. Deepchecks: A Library for Testing and Validating Machine Learning Models and Data. *arXiv [cs.LG]*:1-7. https://doi.org/10.48550/arXiv.2203.08491

Cobo, M.J., López-Herrera, A.G., Herrera-Viedma, E. & Herrera, F. 2011. Science Mapping Software Tools: Review, Analysis, and Cooperative Study Among Tools. *Journal of the American Society for Information Science and Technology*, 62(7):1382-1402. https://doi.org/10.1002/asi.21525 10.1002/asi.21525

Das, M., Cui, R., Campbell, D.R., Agrawal, G. & Ramnath, R. 2015. *Towards methods for systematic research on big data*. Paper presented at the 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA. https://doi.org/10.1109/BigData.2015.7363989 Date of access: 7 September 2023.

Donald, G. 2018. A brief summary of pilot and feasibility studies: Exploring terminology, aims, and methods. *European Journal of Integrative Medicine*, 24:65-70. https://doi.org/10.1016/j.eujim.2018.10.017 10.1016/j.eujim.2018.10.017

Dorogush, A.V., Ershov, V. & Gulin, A. 2018. CatBoost: gradient boosting with categorical features support. *arXiv [cs.LG]*:1-7. https://doi.org/10.48550/arXiv.1810.11363

Doshi-Velez, F. & Kim, B. 2017. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv [stat.ML]*:1-13. https://doi.org/10.48550/arXiv.1702.08608

Eastwood, C., Klerkx, L., Ayre, M. & Dela Rue, B. 2019. Managing Socio-Ethical Challenges in the Development of Smart Farming: From a Fragmented to a Comprehensive Approach for Responsible Research and Innovation. *Journal of Agricultural & Environmental Ethics*, 32:741-768. https://doi.org/10.1007/s10806-017-9704-5 10.1007/s10806-017-9704-5

Ehsan, U., Wintersberger, P., Liao, Q.V., Mara, M., Streit, M., Wachter, S., ... Riedl, M.O. 2021. *Operationalizing Human-Centered Perspectives in Explainable AI*. Paper presented at the In Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems (CHI EA '21), New York, NY, USA. https://doi.org/10.1145/3411763.3441342 Date of access: 7 September 2023.

Elahi, E., Khalid, Z., Tauni, M.Z., Zhang, H.X. & Xing, L.R. 2022. Extreme weather events risk to crop-production and the adaptation of innovative management strategies to mitigate the risk: A retrospective survey of rural Punjab, Pakistan. *Technovation*, 117:1-12. https://doi.org/10.1016/j.technovation.2021.102255 10.1016/j.technovation.2021.102255

Freund, Y. & Schapire, R.E. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119-139. https://doi.org/10.1006/jcss.1997.1504 10.1006/jcss.1997.1504

Garfield, E. 1990. Keywords Plus - Isi's Breakthrough Retrieval Method .1. Expanding Your Searching Power on Current-Contents on Diskette. *Current Contents*, 32:5-9.

Garouani, M., Ahmad, A., Bouneffa, M., Hamlich, M., Bourguin, G. & Lewandowski, A. 2022. Towards big industrial data mining through explainable automated machine learning. *International Journal of Advanced Manufacturing Technology*, 120:1169-1188. https://doi.org/10.1007/s00170-022-08761-9 10.1007/s00170-022-08761-9

Gashi, M., Mutlu, B. & Thalmann, S. 2023. Impact of Interdependencies: Multi-Component System Perspective toward Predictive Maintenance Based on Machine Learning and XAI. *Applied Sciences-Basel*, 13(5):1-17. https://doi.org/10.3390/app13053088 10.3390/app13053088

Ghasemkhani, B., Aktas, O. & Birant, D. 2023. Balanced K-Star: An Explainable Machine Learning Method for Internet-of-Things-Enabled Predictive Maintenance in Manufacturing. *Machines*, 11(3):1-20. https://doi.org/10.3390/machines11030322 10.3390/machines11030322

Goel, V. & Dolan, R.J. 2003. Explaining modulation of reasoning by belief. *Cognition*, 87(1):B11-B22. https://doi.org/10.1016/s0010-0277(02)00185-3 10.1016/s0010-0277(02)00185-3

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F. & Pedreschi, D. 2018. A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys*, 51(5):1-42. https://doi.org/10.1145/3236009 10.1145/3236009

Hajgató, G., Wéber, R., Szilágyi, B., Tóthpál, B., Gyires-Tóth, B. & Hős, C. 2022. PredMaX: Predictive maintenance with explainable deep convolutional autoencoders. *Advanced Engineering Informatics*, 54:1-9. https://doi.org/10.1016/j.aei.2022.101778 10.1016/j.aei.2022.101778

Hand, D.J. & Till, R.J. 2001. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning*, 45(2):171-186. https://doi.org/10.1023/A:1010920819831 10.1023/A:1010920819831

Harmani, V.P., Himawan, B.M., Alhadi, M.A., Gunawan, A.A.S. & Anderies. 2022. *Systematic Literature Review: Implementation Of Artificial Intelligence in Precision Agriculture*. Paper presented at the 2022 5th International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia. https://doi.org/10.1109/ICOIACT55506.2022.9971917 Date of access: 7 September 2023.

Hastie, T., Rosset, S., Zhu, J. & Zou, H. 2009. Multi-class AdaBoost. *Statistics and Its Interface*, 2(3):349-360. https://dx.doi.org/10.4310/SII.2009.v2.n3.a8 10.4310/SII.2009.v2.n3.a8

Hepenstal, S., Zhang, L. & Wong, B.L.W. 2021. *An analysis of expertise in intelligence analysis to support the design of Human-Centered Artificial Intelligence*. Paper presented at the 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Melbourne, Australia. https://doi.org/10.1109/SMC52423.2021.9659095 Date of access: 7 September 2023.

Hermansa, M., Kozielski, M., Michalak, M., Szczyrba, K., Wrobel, L. & Sikora, M. 2021. Sensor-Based Predictive Maintenance with Reduction of False Alarms-A Case Study in Heavy Industry. *Sensors (Basel)*, 22(1):1-26. https://doi.org/10.3390/s22010226 10.3390/s22010226

Hevner, A.R., March, S.T., Park, J. & Ram, S. 2004. Design science in Information Systems research. *Mis Quarterly*, 28(1):75-105. https://doi.org/10.2307/25148625 10.2307/25148625

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., ... Kingsbury, B. 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, 29(6):82-97. https://doi.org/10.1109/MSP.2012.2205597 10.1109/msp.2012.2205597

Ho, T. 1998. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832-844. https://doi.org/10.1109/34.709601 10.1109/34.709601

Hochreiter, S. & Schmidhuber, J. 1997. Long short-term memory. *Neural Comput*, 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 Date of access: 7 September 2023. 10.1162/neco.1997.9.8.1735

Huang, G.B., Zhu, Q.Y. & Siew, C.K. 2006. Extreme learning machine: Theory and applications. *Neurocomputing*, 70:489-501. https://doi.org/10.1016/j.neucom.2005.12.126 10.1016/j.neucom.2005.12.126

Jakubowski, J., Stanisz, P., Bobek, S. & Nalepa, G.J. 2021. Anomaly Detection in Asset Degradation Process Using Variational Autoencoder and Explanations. *Sensors (Basel)*, 22(1):1-20. https://doi.org/10.3390/s22010291 10.3390/s22010291

Jiang, H., Kim, B., Guan, M.Y. & Gupta, M. 2018. To Trust Or Not To Trust A Classifier. *arXiv [stat.ML]*:1-25. https://doi.org/10.48550/arXiv.1805.11783

Kamalov, F. & Sulieman, H. 2021. *Time series signal recovery methods: comparative study*. Paper presented at the 2021 International Symposium on Networks, Computers and Communications (ISNCC), Dubai, United Arab Emirates. https://doi.org/10.1109/ISNCC52172.2021.9615669 Date of access: 7 September 2023.

Kande, M., Isaksson, A.J., Thottappillil, R. & Taylor, N. 2017. Rotating Electrical Machine Condition Monitoring Automation—A Review. *Machines*, 5(4):1-15. https://doi.org/10.3390/machines5040024 10.3390/machines5040024

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. 2017. *LightGBM: a highly efficient gradient boosting decision tree*. Paper presented at the In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Red Hook, NY, USA. https://dl.acm.org/doi/10.5555/3294996.3295074 Date of access: 7 September 2023.

Keany, E. 2023. *BorutaShap*. https://github.com/Ekeany/Boruta-Shap Date of access: 7 September 2023.

Khan, S.H., Hayat, M., Bennamoun, M., Sohel, F.A. & Togneri, R. 2018. Cost-Sensitive Learning of Deep Feature Representations From Imbalanced Data. *IEEE Trans Neural Netw Learn Syst*, 29(8):3573-3587. https://doi.org/10.1109/TNNLS.2017.2732482 10.1109/TNNLS.2017.2732482

Kononov, E., Klyuev, A. & Tashkinov, M. 2023. Prediction of Technical State of Mechanical Systems Based on Interpretive Neural Network Model. *Sensors (Basel)*, 23(4):1-19. https://doi.org/10.3390/s23041892 10.3390/s23041892

Krizhevsky, A., Sutskever, I. & Hinton, G.E. 2017. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM*, 60(6):84-90. https://dl.acm.org/doi/10.1145/3065386 Date of access: 7 September 2023. 10.1145/3065386

Kursa, M.B., Jankowski, A. & Rudnicki, W.R. 2010. Boruta - A System for Feature Selection. *Fundamenta Informaticae*, 101(4):271-286. http://dx.doi.org/10.3233/FI-2010-288 10.3233/Fi-2010-288

Kuzlu, M., Cali, U., Sharma, V. & Güler, Ö. 2020. Gaining Insight Into Solar Photovoltaic Power Generation Forecasting Utilizing Explainable Artificial Intelligence Tools. *IEEE Access*, 8:187814-187823. https://doi.org/10.1109/ACCESS.2020.3031477 10.1109/Access.2020.3031477

Lane, D., Hebl, M., Guerra, R., Osherson, D. & Zimmer, H. 2003. *Introduction to Statistics*.

Langone, R., Cuzzocrea, A. & Skantzos, N. 2020. Interpretable Anomaly Prediction: Predicting anomalous behavior in industry 4.0 settings via regularized logistic regression tools. *Data & Knowledge Engineering*, 130:1-20. https://doi.org/10.1016/j.datak.2020.101850 10.1016/j.datak.2020.101850

Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278-2324. https://doi.org/10.1109/5.726791 10.1109/5.726791

López, V., Fernández, A., García, S., Palade, V. & Herrera, F. 2013. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113-141. https://doi.org/10.1016/j.ins.2013.07.007 10.1016/j.ins.2013.07.007

Louppe, G. & Geurts, P. 2012. *Ensembles on Random Patches*. Paper presented at the Machine Learning and Knowledge Discovery in Databases, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33460-3_28 Date of access: 7 September 2023.

Luengo, J., García-Gil, D., Ramírez-Gallego, S., García, S. & Herrera, F. 2020. *Big Data Preprocessing*. Springer International Publishing.

Lughofer, E. & Sayed-Mouchaweh, M. 2019. Predictive Maintenance In Dynamic Systems. 1:311-425. http://dx.doi.org/10.1007/978-3-030-05645-2 10.1007/978-3-030-05645-2

Lundberg, S.M. & Lee, S.-I. 2017. *A unified approach to interpreting model predictions*. Paper presented at the In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Red Hook, NY, USA. https://dl.acm.org/doi/10.5555/3295222.3295230 Date of access: 7 September 2023.

Martinez, I., Viles, E. & Olaizola, I.G. 2021. Data Science Methodologies: Current Challenges and Future Approaches. *Big Data Research*, 24:1-18. https://doi.org/10.1016/j.bdr.2020.100183 10.1016/j.bdr.2020.100183

McClish, D.K. 1989. Analyzing a portion of the ROC curve. *Med Decis Making*, 9(3):190-195. https://doi.org/10.1177/0272989X8900900307 10.1177/0272989X8900900307

McCulloch, W.S. & Pitts, W. 1943. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115-133. https://doi.org/10.1007/BF02478259 10.1007/bf02478259

McKenna, S., Richardson, J. & Manroop, L. 2011. Alternative paradigms and the study and practice of performance management and evaluation. *Human Resource Management Review*, 21(2):148-157. https://doi.org/10.1016/j.hrmr.2010.09.002 10.1016/j.hrmr.2010.09.002

Mey, O. & Neufeld, D. 2022. Explainable AI Algorithms for Vibration Data-Based Fault Detection: Use Case-Adadpted Methods and Critical Evaluation. *Sensors (Basel)*, 22(23):1-22. https://doi.org/10.3390/s22239037 10.3390/s22239037

Microsoft. 2021. *Telemetry for Predictive Maintenance*. https://www.kaggle.com/datasets/arnabbiswas1/microsoft-azure-predictive-maintenance Date of access: 23 March 2023.

Minaee, S., Kafieh, R., Sonka, M., Yazdani, S. & Jamalipour Soufi, G. 2020. Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning. *Med Image Anal*, 65:1-9. https://doi.org/10.1016/j.media.2020.101794 10.1016/j.media.2020.101794

Molnar, C. 2022. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. 2nd ed.

Molnar, C., Casalicchio, G. & Bischl, B. 2020. *Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges*. Springer, Cham. (ECML PKDD 2020 Workshops).

Mothilal, R.K., Sharma, A. & Tan, C. 2020. *Explaining machine learning classifiers through diverse counterfactual explanations*. Paper presented at the In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT* '20), New York, NY, USA. https://doi.org/10.1145/3351095.3372850 Date of access: 7 September 2023.

Orn, D., Duan, L., Liang, Y., Siy, H. & Subramaniam, M. 2020. *Agro-AI Education: Artificial Intelligence for Future Farmers*. Paper presented at the Proceedings of the 21st Annual Conference on Information

Technology Education, New York, NY, USA. https://doi.org/10.1145/3368308.3415457 Date of access: 7 September 2023.

Page, M.J., McKenzie, J.E., Bossuyt, P.M., Boutron, I., Hoffmann, T.C., Mulrow, C.D., ... Moher, D. 2021. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *Syst Rev*, 10(1):1-11. https://doi.org/10.1186/s13643-021-01626-4 10.1186/s13643-021-01626-4

Panchbhaiyye, V. & Ogunfunmi, T. 2018. *Experimental Results on Using Deep Learning to Identify Agricultural Pests*. Paper presented at the 2018 IEEE Global Humanitarian Technology Conference (GHTC), San Jose, CA, USA. https://doi.org/10.1109/GHTC.2018.8601896 Date of access: 7 September 2023.

Pons, P. & Latapy, M. 2006. Computing Communities in Large Networks Using Random Walks. *Journal of Graph Algorithms and Applications*, 10(2):191-218. https://doi.org/10.7155/jgaa.00124 10.7155/jgaa.00124

Poppe, K., Wolfert, S., Verdouw, C. & Renwick, A. 2015. A European Perspective on the Economics of Big Data. *Farm Policy Journal*, 12(1):11-19. https://www.researchgate.net/publication/278300518_A_European_Perspective_on_the_Economics_of_Big_Data Date of access: 7 September 2023.

Posit. 2023. *Posit*. https://posit.co/downloads Date of access: 28 June 2023.

Pump. 2019. *Pump Sensor Data*. https://www.kaggle.com/datasets/nphantawee/pump-sensor-data Date of access: 23 March 2023.

Rahman, S.A.Z., Mitra, K.C. & Islam, S.M.M. 2018. *Soil Classification Using Machine Learning Methods and Crop Suggestion Based on Soil Series*. Paper presented at the 2018 21st International Conference of Computer and Information Technology (ICCIT), Dhaka, Bangladesh. https://doi.org/10.1109/ICCITECHN.2018.8631943 Date of access: 7 September 2023.

Rana, J. & Paul, J. 2017. Consumer behavior and purchase intention for organic food: A review and research agenda. *Journal of Retailing and Consumer Services*, 38:157-165. https://doi.org/10.1016/j.jretconser.2017.06.004 10.1016/j.jretconser.2017.06.004

Reif, M., Shafait, F., Goldstein, M., Breuel, T. & Dengel, A. 2014. Automatic classifier selection for non-experts. *Pattern Analysis and Applications*, 17(1):83-96. 10.1007/s10044-012-0280-z

Riedl, M.O. 2019. Human-centered artificial intelligence and machine learning. *Human Behavior and Emerging Technologies*, 1(1):33-36. https://doi.org/10.1002/hbe2.117 10.1002/hbe2.117

Rosenblatt, F. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol Rev*, 65(6):386-408. https://doi.org/10.1037/h0042519 10.1037/h0042519

Russell, S.J. & Norvig, P. 2020. *Artificial intelligence : a modern approach*. 4th ed. Boston: Pearson.

Schuster, M. & Paliwal, K.K. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673-2681. 10.1109/78.650093

Selbst, A.D. & Barocas, S. 2018. The Intuitive Appeal of Explainable Machines. *Fordham Law Review*, 87(3):1085-1139. http://dx.doi.org/10.2139/ssrn.3126971 10.2139/ssrn.3126971

Serradilla, O., Zugasti, E., de Okariz, J.R., Rodriguez, J. & Zurutuza, U. 2021. Adaptable and Explainable Predictive Maintenance: Semi-Supervised Deep Learning for Anomaly Detection and Diagnosis in Press Machine Data. *Applied Sciences-Basel*, 11(16):1-20. https://doi.org/10.3390/app11167376 10.3390/app11167376

Shankar, P., Werner, N., Selinger, S. & Janssen, O. 2020. *Artificial Intelligence Driven Crop Protection Optimization for Sustainable Agriculture*. Paper presented at the 2020 IEEE / ITU International Conference

on Artificial Intelligence for Good (AI4G), Geneva, Switzerland. https://doi.org/10.1109/AI4G50087.2020.9311082 Date of access: 7 September 2023.

Shapley, L.S. 1952. *A Value for N-Person Games*. Santa Monica, CA: RAND Corporation.

Shi, X.J., Chen, Z.R., Wang, H., Yeung, D.Y., Wong, W.K. & Woo, W.C. 2015. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *arXiv [cs.CV]*:1-12. https://doi.org/10.48550/arXiv.1506.04214

Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., ... Summers, R.M. 2016. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Trans Med Imaging*, 35(5):1285-1298. https://doi.org/10.1109/tmi.2016.2528162 10.1109/TMI.2016.2528162

Sonka, S. 2015. Big Data: From Hype to Agricultural Tool. *Farm Policy Journal*, 12:1-9. https://www.researchgate.net/publication/279771638_Big_Data_From_Hype_to_Agricultural_Tool Date of access: 23 March 2023.

Sperrle, F., El-Assady, M., Guo, G., Borgo, R., Chau, D.H., Endert, A. & Keim, D. 2021. A Survey of Human-Centered Evaluations in Human-Centered Machine Learning. *Computer Graphics Forum*, 40(3):543-567. https://doi.org/10.1111/cgf.14329 10.1111/cgf.14329

Steurtewagen, B. & Van den Poel, D. 2021. Adding interpretability to predictive maintenance by machine learning on sensor data. *Computers & Chemical Engineering*, 152:1-8. https://doi.org/10.1016/j.compchemeng.2021.107381 10.1016/j.compchemeng.2021.107381

Taravatrooy, N., Nikoo, M.R., Hobbi, S., Sadegh, M. & Izady, A. 2020. A novel hybrid entropy-clustering approach for optimal placement of pressure sensors for leakage detection in water distribution systems under uncertainty. *Urban Water Journal*, 17(3):185-198. https://doi.org/10.1080/1573062X.2020.1758162 10.1080/1573062x.2020.1758162

Turyahikayo, E. 2021. Philosophical paradigms as the bases for knowledge management research and practice. *Knowledge Management & E-Learning-an International Journal*, 13(2):209-224. https://doi.org/10.34105/j.kmel.2021.13.012 10.34105/j.kmel.2021.13.012

Upasane, S.J., Hagras, H., Anisi, M.H., Savill, S., Taylor, I. & Manousakis, K. 2021. *A Big Bang-Big Crunch Type-2 Fuzzy Logic System for Explainable Predictive Maintenance*. Paper presented at the 2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Luxembourg, Luxembourg. https://doi.org/10.1109/FUZZ45933.2021.9494540 Date of access: 7 September 2023.

Vincent, D.R., Deepa, N., Elavarasan, D., Srinivasan, K., Chauhdary, S.H. & Iwendi, C. 2019. Sensors Driven AI-Based Agriculture Recommendation Model for Assessing Land Suitability. *Sensors (Basel)*, 19(17):1-16. https://doi.org/10.3390/s19173667 10.3390/s19173667

Wachter, S., Mittelstadt, B. & Russell, C. 2018. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *arXiv [cs.AI]*:1-52. https://doi.org/10.48550/arXiv.1711.00399

Wang, S. & Yao, X. 2013. Using Class Imbalance Learning for Software Defect Prediction. *IEEE Transactions on Reliability*, 62(2):434-443. https://doi.org/10.1109/TR.2013.2259203 10.1109/Tr.2013.2259203

Wang, S.H. & Chen, S.N. 2019. Insights to fracture stimulation design in unconventional reservoirs based on machine learning modeling. *Journal of Petroleum Science and Engineering*, 174:682-695. https://doi.org/10.1016/j.petrol.2018.11.076 10.1016/j.petrol.2018.11.076

Wijesekara, L. & Liyanage, L. 2021. *Air quality data pre-processing: A novel algorithm to impute missing values in univariate time series*. Paper presented at the 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI), Washington, DC, USA. https://doi.org/10.1109/ICTAI52525.2021.00159 Date of access: 23 March 2023.

Wilhelm, K. 2008. *Pearson's Correlation Coefficient*. 1st ed. Dordrecht: Springer Netherlands. (Encyclopedia of Public Health).

Wolfert, S., Ge, L., Verdouw, C. & Bogaardt, M.J. 2017. Big Data in Smart Farming - A review. *Agricultural Systems*, 153:69-80. https://doi.org/10.1016/j.agsy.2017.01.023 10.1016/j.agsy.2017.01.023

Wu, H., Huang, A. & Sutherland, J.W. 2021. Layer-wise relevance propagation for interpreting LSTM-RNN decisions in predictive maintenance. *The International Journal of Advanced Manufacturing Technology*, 118:963-978. https://doi.org/10.1007/s00170-021-07911-9 10.1007/s00170-021-07911-9

Yildirim, M., Gebraeel, N.Z. & Sun, X.A. 2017. Integrated Predictive Analytics and Optimization for Opportunistic Maintenance and Operations in Wind Farms. *IEEE Transactions on Power Systems*, 32(6):4319-4328. https://doi.org/10.1109/TPWRS.2017.2666722 10.1109/Tpwrs.2017.2666722

Zeiler, M.D. & Fergus, R. 2013. Visualizing and Understanding Convolutional Networks. *arXiv [cs.CV]*:1-11. https://doi.org/10.48550/arXiv.1311.2901

Zhong, Y., Lai, I.K., Guo, F. & Tang, H. 2021. Research on Government Subsidy Strategies for the Development of Agricultural Products E-Commerce. *Agriculture*, 11(11):1-28. https://doi.org/10.3390/agriculture11111152 10.3390/agriculture11111152

Zhou, P. & Yin, P.T. 2019. An opportunistic condition-based maintenance strategy for offshore wind farm based on predictive analytics. *Renewable & Sustainable Energy Reviews*, 109:1-9. https://doi.org/10.1016/j.rser.2019.03.049 10.1016/j.rser.2019.03.049

APPENDICES

Appendix A – Additional Tables and Figures

Table A.1: Managing synonyms

Used	Replaced		
artificial intelligence	ai		
machine learning	ml		
explainable artificial intelligence	xai	explainable ai	explainable ai (xai)
interpretable artificial intelligence	iai	interpretable ai	interpretable ai (iai)
interpretable machine learning	iml	interpretable ml	
artificial intelligence	artificial intelligence (ai)		
machine learning	machine learning (ml)		
explainable artificial intelligence	explainable artificial intelligence (xai)		
interpretable artificial intelligence	interpretable artificial intelligence (iai)		
interpretable machine learning	interpretable machine learning (iml)		

Appendix B – Code and Tools

Housed in this GitHub repository: https://github.com/iammelvink, one discovers Python code, meticulously made for experimental execution; this code, a tangible manifestation of algorithms dissected in this dissertation, fosters reproducibility.

Table B.1: Supporting Python libraries

Library	Version
pandas	1.5.3
numpy	1.23.5
matplotlib	3.7.1
seaborn	0.11.0

tensorflow	2.10.1
keras	2.10.0
sklearn	1.3.0
xgboost	2.0.0
shap	0.42.1
dice_ml	0.10

Table B.2: DL - Convolutional Long-Short-Term Memory Neural Network parameters (Telemetry for Predictive Maintenance)

Layer (type)	Output Shape	Number of Params	Definition
batch_normalisation	(None, 19)	76	Normalises and scales input data, improving training efficiency
reshape (Reshape)	(None, 19, 1, 1)	0	Reorganises the input into a (19, 1, 1) shape without additional parameters.
conv_lstm1d (ConvLSTM1D)	(None, 19, 1, 4)	96	Applies 1D Convolutional Long Short-Term Memory (LSTM) with four output channels
flatten (Flatten)	(None, 76)	0	Converts the input into a flat vector, keeping the shape unchanged
dropout (layer)	(None, 76)	0	It helps regularise the model by randomly dropping a percentage of inputs
dense (Dense)	(None, 5)	385	A fully connected layer with five output units

Parameter	Value	Definition
activation	“relu”	Rectified Linear Unit (ReLU) activation function
l1 regulariser	0.001	L1 regularisation term applied to the model's weights
l2 regulariser	0.001	L2 regularisation term applied to the model's weights
Dropout (helps to regularise)	0.1	Dropout applied during training to prevent overfitting

optimisers	Adam	Adam optimiser for gradient-based optimisation
learning_rate	0.002	The rate at which the model's weights are updated during training
loss	"sparse_categorical_crossentropy"	Loss function used to measure model performance
EarlyStopping	TRUE	A technique to stop training when a specific condition is met
monitor	“val_loss”	The metric being monitored during early stopping
patience	25	The number of epochs to wait before triggering early stopping
epoch	276	The number of times the model goes through the training dataset
kernel_size	1	The size of the convolutional kernel used in the model

Table B.3.1: ML - Bagging classifier parameters (Telemetry for Predictive Maintenance)

Parameter	Value	Definition
class_weight	'balanced'	It assigns weights to classes to balance the model's performance
criterion	'entropy'	Uses the entropy criterion for Decision Tree node splitting
max_depth	9	Specifies the maximum depth of the Decision Tree
n_estimators	100	Specifies the number of Decision Trees (estimators) in the ensemble as 100
n_jobs	-1	To maximise processor cores (-1) for parallel processing to speed up model training

Table B.3.2: ML - Decision Tree classifier parameters (Telemetry for Predictive Maintenance)

Parameter	Value	Definition
class_weight	'balanced'	It assigns weights to classes to balance the model's performance
criterion	'entropy'	Uses the entropy criterion for Decision Tree node splitting

max_depth	9	Specifies the maximum depth of the Decision Tree

Table B.4: DL - Bidirectional Recurrent Neural Network and Long-Short-Term Memory Neural Network parameters (Pump Sensor)

Layer (type)	Output Shape	Number of Params	Definition
batch_normalisation (BatchNormalisation)	(None, 4)	16	Normalises and scales input data, improving training efficiency
reshape (Reshape)	(None, 1, 4)	0	Rearranges input into a (1, 4) shape with no additional parameters.
bidirectional (Bidirectional)	(None, 1, 16)	832	It uses bidirectional recurrent layers with 16 output channels
batch_normalisation_1 (BatchNormalisation)	(None, 1, 16)	64	Normalises and scales the output of the previous dense layer
bidirectional_1 (Bidirectional)	(None, 2)	144	Employs bidirectional recurrent layers with two output channels
dropout (layer)	(None, 2)	0	It helps regularise the model by randomly dropping a percentage of inputs
dense (Dense)	(None, 3)	9	A fully connected layer with three output units
Parameter		**Value**	**Definition**
activation		“tanh”	Hyperbolic Tangent (tanh) activation function
l1 regulariser		0.001	L1 regularisation term applied to the model's weights
l2 regulariser		0.001	L2 regularisation term applied to the model's weights

Dropout (helps to regularise)	0.1	Dropout applied during training to prevent overfitting
optimisers	Adam	Adam optimiser for gradient-based optimisation
learning_rate	0.02	The rate at which the model's weights are updated during training
loss	"sparse_categorical_crossentropy"	Loss function used to measure model performance
EarlyStopping	TRUE	A technique to stop training when a specific condition is met
monitor	“val_loss”	The metric being monitored during early stopping
patience	30	The number of epochs to wait before triggering early stopping
epoch	246	The number of times the model goes through the training dataset

Table B.5: ML - Adaptive Boosting classifier parameters (Pump Sensor)

Parameter	Value	Definition
learning_rate	0.2	Sets the learning rate to 0.2, determining the step size for model updates
n_estimators	100	Defines the number of estimators (weak models) to be included in the Adaptive Boosting classifier as 100

Appendix C – Research Publications

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 January 2024, accepted 6 February 2024, date of publication 13 February 2024, date of current version 20 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3365586

RESEARCH ARTICLE

Explainable Artificial Intelligence Model for Predictive Maintenance in Smart Agricultural Facilities

MELVIN KISTEN[1], ABSALOM EL-SHAMIR EZUGWU[1], AND MICHEAL O. OLUSANYA[2], (Member, IEEE)
[1]Unit for Data Science and Computing, North-West University, Potchefstroom 2520, South Africa
[2]Department of Computer Science and Information Technology, Sol Plaatje University, Kimberley 8300, South Africa
Corresponding author: Absalom El-Shamir Ezugwu (absalom.ezugwu@nwu.ac.za)

ABSTRACT Artificial Intelligence (AI) in Smart Agricultural Facilities (SAF) often lacks explainability, hindering farmers from taking full advantage of their capabilities. This study tackles this gap by introducing a model that combines eXplainable Artificial Intelligence (XAI), with Predictive Maintenance (PdM). The model aims to provide both predictive insights and explanations across four key dimensions, namely data, model, outcome, and end-user. This approach marks a shift in agricultural AI, reshaping how these technologies are understood and applied. The model outperforms related studies, showing quantifiable improvements. Specifically, the Long-Short-Term Memory (LSTM) classifier shows a 5.81% rise in accuracy. The eXtreme Gradient Boosting (XGBoost) classifier exhibits a 7.09% higher F1 score, 10.66% increased accuracy, and a 4.29% increase in Receiver Operating Characteristic-Area Under the Curve (ROC-AUC). These results could lead to more precise maintenance predictions in real-world settings. This study also provides insights into data purity, global and local explanations, and counterfactual scenarios for PdM in SAF. It advances AI by emphasising the importance of explainability beyond traditional accuracy metrics. The results confirm the superiority of the proposed model, marking a significant contribution to PdM in SAF. Moreover, this study promotes the understanding of AI in agriculture, emphasising explainability dimensions. Future research directions are advocated, including multi-modal data integration and implementing Human-in-the-Loop (HITL) systems aimed at improving the effectiveness of AI and addressing ethical concerns such as Fairness, Accountability, and Transparency (FAT) in agricultural AI applications.

INDEX TERMS Agriculture, smart agricultural facilities, predictive maintenance, machine learning, deep learning, explainable artificial intelligence.

LIST OF ABBREVIATIONS

Abbreviation	Definition
AI	Artificial Intelligence.
ANN	Artificial Neural Network.
AUC	Area Under the Curve.
CFE	Counterfactual Explanations.
CNN	Convolutional Neural Network.
CPU	Central Processing Unit.
CSIR	Council for Scientific and Industrial Research.
CSML	Cost-Sensitive Machine Learning.
DiCE	Diverse Counterfactual Explanations.
DL	Deep Learning.
DNN	Deep Neural Network.
DT	Decision Tree.
EL	Ensemble Learning.
ELI5	Explain Like I Am Five.
ELM	Extreme Learning Machine.
FAT	Fairness Accountability Transparency.
FN	False Negative.
FP	False Positive.
GB	Gradient Boosting.
GBDT	Gradient Boosting Decision Tree.
GBM	Gradient Boosting Model.
GDPR	General Data Protection Regulation.
GHz	Gigahertz.

The associate editor coordinating the review of this manuscript and approving it for publication was Tyson Brooks.

GPU	Graphics Processing Unit.
HITL	Human-in-the-Loop.
IAI	Interpretable Artificial Intelligence.
ID	Identifier.
LGBM	Light Gradient Boosting Model.
LIME	Local Interpretable Model-agnostic Explanations.
LOCF	Last Observation Carried Forward.
LR	Linear Regression.
LRP	Layer-wise Relevance Propagation.
LSTM	Long-Short-Term Memory.
LVQ	Learning Vector Quantisation.
MB	Megabyte.
ML	Machine Learning.
MLP	Multi-Layer Perceptron.
NN	Neural Network.
OC	One-Class.
OC-SVM	One-Class Support Vector Machine.
OS	Operating System.
PdM	Predictive Maintenance.
PPS	Predictive Power Score.
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses.
RAM	Random-Access Memory.
ReLU	Rectified Linear Unit.
RF	Random Forest.
RFECV	Recursive Feature Elimination with Cross-Validation.
RNN	Recurrent Neural Network.
ROC	Receiver Operating Characteristic.
ROC-AUC	Receiver Operating Characteristic-Area Under the Curve.
RTX	Ray Tracing Texel eXtreme.
RUL	Remaining Useful Life.
SAF	Smart Agricultural Facilities.
SHAP	SHapley Additive exPlanations.
SLR	Systematic Literature Review.
SRBD	Systematic Research on Big Data.
STD	Standard Deviation.
SVM	Support Vector Machine.
TANH	Hyperbolic Tangent.
TL	Transfer Learning.
TM	Trademark.
TN	True Negative.
TP	True Positive.
USA	United States of America.
VT	Variance Thresholding.
WoS	Web of Science.
XAI	eXplainable Artificial Intelligence.
XGB	eXtreme Gradient Boosting.

I. INTRODUCTION

Agricultural enterprises grapple with maintenance complexities and costly machinery [1], [2], [3], [4], [5], [6], [7]. These challenges threaten equipment upkeep and diminish productivity. Responding, farmers adopt Smart Agricultural Facilities (SAF), transforming farming practices with technologies like sensors and drones [8], [9]. These innovations not only streamline tasks such as soil classification [10], pest management [11], and water leakage detection [12], [13], but also enhance sustainability and efficiency. However, SAF systems, while transformative, are prone to malfunctions. The Predictive Maintenance (PdM) literature offers strategies to mitigate these problems, promoting operational efficiency, and reducing costs [14], thus sustaining and optimising SAF.

PdM in agriculture transcends mere data collection, revolutionising equipment upkeep. Farmers predict machine failures by leveraging data, enhancing efficiency, slashing costs, and boosting output [15]. However, the success of PdM and SAF is based on seamless data integration, user acceptance of new technologies, and strict privacy adherence [16], [17]; these factors determine the viability of these innovations in agriculture. With challenges in PdM and SAF, eXplainable Artificial Intelligence (XAI) becomes critical, especially under legal mandates like the General Data Protection Regulation (GDPR), demanding explainability in automated systems [18], [19]; this highlights the need for practical, user-friendly, and legally compliant AI models [20]; this study probes a fundamental question: How does one trust AI predictions in SAF without explanations? It explores XAI model development, focusing on PdM strategies [21] and AI explainability methods [22].

Recent AI models for PdM in SAF often obscure their decision-making process, a significant shortfall [23]; this opaqueness, exacerbated by literature focusing narrowly on accuracy and F1 scores without detailed train and test results, diminishes transparency and robustness. Additionally, AI's role in agriculture, as either a farmer's aid or replacement, remains unclear [24]. Consequently, farmers find current AI models in agriculture challenging to grasp, limiting their utility; this gap calls for XAI to improve the understanding and usability of AI in PdM and SAF [25], [26], [27], [28]. Therefore, this study proposes a model blending XAI with PdM in SAF to improve transparency and practicality.

Although some studies on AI explainability adopt a broad, universal approach [29], this study seeks a more detailed, flexible explainability method, tailored to varied stakeholder needs. It focuses on matching explanation depth to the unique needs of diverse users, like AI experts and farmers; this approach recognises that explanation purposes differ, requiring a more customised framework. Specifically, this study highlights the importance of explanations that provide scientific insight and meet stakeholders' internal needs, aiming for a more refined use of AI in agricultural PdM.

Embarking on this study involves conducting a thorough Systematic Literature Review (SLR) with bibliometric analysis to understand the current state of AI models, honing in on XAI. It involved delving into the literature on Deep Learning (DL) and Machine Learning (ML) in PdM, emphasising

XAI's importance due to the opaque decision-making of AI models in SAF [23]. These explorations shape the study's objectives and influence the development of a combined model, highlighting XAI's role in agricultural PdM. The study evaluates four DL and seven ML algorithms, selecting the top-performing ones based on the test F1 score for explainability and for comparative analysis. The research aims to (1) predict maintenance needs and (2) provide explanations using XAI in SAF. To this end, the study focuses on: (i) Assessing AI models' status and the need for explainability through SLR and bibliometric analysis. (ii) Develop DL and ML models for PdM. (iii) Evaluation of the effectiveness of these models. (iv) Elucidating the rationale behind the AI model's predictions. The key contributions of this study are summarised as follows:

1. It pioneers integrating XAI with PdM in SAF, focusing on four explainability dimensions: (1) data, (2) model, (3) outcome, and (4) end-user.
2. Conducts an SLR on the current state of AI models, highlighting the necessity of XAI.
3. Presents transparent training and testing outcomes using the more stringent Receiver Operating Characteristic-Area Under the Curve (ROC-AUC) metric, underscoring the importance of XAI beyond just accuracy.
4. Adapts design principles from XAI and PdM for the specific context of SAF.
5. Strengthens the theoretical underpinnings and future trajectories of XAI in PdM for SAF.

II. LITERATURE REVIEW

This study undertook an extensive literature review to analyse advanced AI-driven PdM techniques critically. It aimed to identify and assess XAI models for PdM, focusing on their role in enhancing maintenance practices. The study also aimed to shed light on the strengths, limitations, and practicality of various XAI models in PdM. Evaluating and synthesising these findings aimed to provide a comprehensive overview of the current state and future direction of XAI models in agricultural PdM.

A. PREDICTIVE MAINTENANCE

This review uncovered three PdM approaches: (1) anomaly detection, (2) prognostics, and (3) diagnostics [21]. Anomaly detection is akin to spotting the unusual in data. Prognostics predict future system performance, while diagnostics identify current issues through performance analysis. Among the reviewed literature, eleven studies focused on prognostics [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], three on anomaly detection [41], [42], [43], and two on both prognostics and diagnostics [40], [44]. Notably, none focused solely on diagnostics, revealing a significant research gap. Future studies should explore how anomaly detection and prognostics can lead to effective diagnostics, thereby improving the robustness and efficiency of PdM in SAF.

B. DEEP AND MACHINE LEARNING FOR PREDICTIVE MAINTENANCE

For prognostics, the Recurrent Neural Networks (RNNs) and Long-Short-Term Memory (LSTM) stand out for their precision; they achieved a notable 90.07% accuracy [33]. Similarly, in predicting Remaining Useful Life (RUL), boundaries were pushed with Bidirectional Recurrent Neural Networks (Bi-RNN) and LSTMs, attaining a 96.15% precision [30]. LSTMs play a pioneering role in anomaly detection, complemented by One-Class Support Vector Machines (OC-SVM), which significantly reduce false alarms [38]. However, OC-SVMs struggle with supervised problems.

An alternative study that used Random Forest (RF) in prognostics also incorporated AutoML. The Random Forest (RF), showed versatility, especially in component-level analysis [32], [36]. However, AutoML's generalist approach, while democratising ML, hinders model optimisation [45]. Ensemble Learning (EL) offered diverse algorithmic solutions and found utility in the prognostics realm of manufacturing industries [36]. Despite their complexity, other methods like Balanced K-Star, Multi-Layer Perceptron (MLP), Extreme Learning Machine (ELM), and Transfer Learning (TL) provide alternative approaches, as do Deep Convolutional AutoEncoders [34], [35], [40], [46]. Diagnostics in PdM have been less explored, with only a few studies touching on it [40]. In general, PdM combines a variety of methodologies, each with unique strengths and challenges, necessitating ongoing critical evaluation.

C. EXPLAINABLE ARTIFICIAL INTELLIGENCE

In the fast-paced era of DL and ML, sophisticated model deployment now pervades sectors like healthcare, finance, and agriculture. However, the intricate nature of these models often clouds their decision-making processes, raising concerns about transparency [23]. This opacity has catalysed the emergence of explainability in DL and ML, a concept that transcends mere transparency. Explainability entails dissecting the complexities of the DL and ML models to render their decision-making understandable, catering to both experts and non-experts. Crucially, explainability embodies a spectrum of facets, each pivotal to demystifying and ensuring the reliability of the DL models.

1) DIMENSIONS OF EXPLAINABILITY

Four explainability dimensions were extracted from the reviewed literature: (1) data, (2) model, (3) outcome, and (4) end-user. The "data dimension" delves into the data's limitations and potential [22]. However, most studies overlooked these aspects, failing to assess whether the data could support the insights sought; this oversight calls for more in-depth research on the data capabilities for PdM in SAF. The "model dimension" explores how input data influences model predictions [22]. Often, researchers assume feature independence, a notion prone to bias. Despite this, most studies incorporated this dimension, with a few focusing

on both model and ''outcome dimensions'' [30], [31], [32], [36], [39]. The review revealed only two studies dedicated to outcome explainability [34], [37], indicating a research gap in understanding the reasoning behind single-instance AI model predictions. Addressing this can enhance transparency and decision-making in AI models. The ''end-user'' dimension, which tailors explanations to non-technical users [47], remained unexplored in the reviewed literature, signalling a need for research that makes AI understandable to a broader audience.

2) APPROACHES TO EXPLAINABILITY

Six explainability approaches emerged from the literature [48]: local explainability, global explainability, model-specific, model-agnostic, model-centric, and data-centric approaches. Local explainability clarifies individual predictions, while global explainability unveils overall model behaviour. Two studies tackled both local and global explainability [33], [41], but none solely focused on global explainability, indicating a research gap. Thirteen studies explored local explainability alone [30], [31], [32], [34], [35], [36], [37], [38], [39], [40], [42], [43], [44].

Model-specific approaches are confined to specific AI models, whereas model-agnostic strategies apply universally. Ten studies used model-agnostic methods [30], [31], [32], [34], [36], [38], [40], [42], [43], [44], while three used model-specific approaches [33], [35], [37], and only two studies harnessed both [39], [41]. Model-centric approaches analyse input-output relationships within models, while data-centric approaches focus on data quality and relevance [47]. All reviewed studies focused on model-centric approaches, leaving data-centric strategies largely unexplored, thus highlighting a significant research opportunity.

D. EXPLAINABLE ARTIFICIAL INTELLIGENCE FOR PREDICTIVE MAINTENANCE

SHapley Additive exPlanations (SHAP) stood out, despite complexities [30], [32], [36], [38], [39], [40], [42], [44]. SHAP was used to clarify feature impacts for false alarm predictions [38]; its diagnostic application helped in model interpretation [40], [44]. Local Interpretable Model-agnostic Explanations (LIME), versatile in various AI models, excelled in specific, localised predictions, as demonstrated in transportation anomaly detection [41]. However, LIME's local focus limits a broader understanding of the model.

Layer-wise Relevance Propagation (LRP), suited for DL, offered detailed insights into prediction influences [33]. LRP's effectiveness was notable, but model-specific. A comparison of LIME, SHAP, and Explain Like I Am Five (ELI5) revealed varied feature attributions and efficiency [39]. LIME was shown to be efficient; ELI5 provided more intuitive explanations, but lacked model-agnostic versatility. Counterfactual Explanations (CFE) have gained popularity for enhancing AI acceptability, especially among non-experts [34], [36]. Consequently, while XAI for PdM presents various tools, challenges in complexity, accessibility, and applicability, these highlight the ongoing quest for a balance between technical depth and user-friendly explanations.

TABLE 1. Databases and search strings.

Database	Search string
Web of Science (WoS)	(TS=(explain* OR interpret* OR xai OR iai) AND TS=(ai OR ml) AND TS=(maintenance OR detect* OR prognos* OR diagnos*)) AND (PY==("2012" OR "2013" OR "2014" OR "2015" OR "2016" OR "2017" OR "2018" OR "2019" OR "2020" OR "2021" OR "2022") AND DT==("ARTICLE") AND LA==("ENGLISH") AND SJ==("COMPUTER SCIENCE"))
Scopus	(TITLE-ABS-KEY (explain* OR interpret* OR xai OR iai) AND TITLE-ABS-KEY (ai OR ml) AND TITLE-ABS-KEY (maintenance OR detect* OR prognos* OR diagnos*)) AND PUBYEAR > 2011 AND PUBYEAR < 2023 AND (LIMIT-TO (SUBJAREA , "COMP")) AND (LIMIT-TO (DOCTYPE , "ar")) AND (LIMIT-TO (LANGUAGE , "English")) AND (LIMIT-TO (SRCTYPE , "j"))

TABLE 2. Search parameters.

Parameter	Value
Timespan	2012-2022
Language	English
Subject area	Computer Science
Document types	Article
Sources	Journals

E. BIBLIOMETRIC ANALYSIS

This study undertook a detailed bibliometric analysis emphasising transparency in AI. The methodology encompassed data processes, descriptive statistics, keyword analysis, knowledge synthesis, and exploration of conceptual and social structures. Searches focused on ''Titles'', ''Abstracts'', and ''Subject Headings'', covering literature from 2012 to 2022, excluding the incomplete year 2023. In the databases, ''Keywords Plus'' and ''Keywords'' served as Subject Headings, enabling a thorough search through synonyms, domain-specific terms, and relevant keywords [49]; this approach helped identify literature pertinent to the research topic. Table 1 details the specific search strings used in the Web of Science (WoS) and Scopus databases. Also, note the parameters selected in the search strings in Table 2.

Fig. 1 depicts a Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) flow diagram [50], outlining the data collection, screening, and analysis steps and how resources were chosen or excluded for the review. The ''Identification'' stage involved gathering 1711 records from two academic databases (WoS and Scopus), with

361 duplicates removed. Screening assessed each record's relevance, leading to the exclusion of 51 preprints and non-journal records, based on criteria in Table 2. The "Included" stage refers to the final selection of 759 records for the review.

1) REVIEW METHODOLOGY

The review involved a rigorous bibliometric analysis, sourcing data from the WoS and Scopus databases on 30 June 2023. A uniform search string on both platforms yielded 759 publications for this review. Data from WoS and Scopus were merged using the Bibiometrix R software package [51], employing the "mergeDbSources" function for integration and the "remove.duplicated" feature to ensure no duplication. Data pre-processing, crucial for accuracy, involved filtering terms like "na", "n/a", "n.a", and "0", and managing synonyms to maintain data consistency [52]. The study conducted a comprehensive bibliometric and critical analysis using RStudio, enhanced by the Bibliometrix package [51], streamlining the review [53].

2) DESCRIPTIVE STATISTICS

The bibliometric analysis included 759 journal articles, with an average age of 2.16 years and an average citation count of 14.24 per document. The yearly expansion rate stood at 44.85%, encompassing 35274 references. The analysis revealed 3878 instances of "Keywords Plus" [49] and 2564 instances of "Author Keywords". Table 3 provides an overview of the final dataset used in this bibliometric analysis.

3) KEYWORD ANALYSIS

Keywords served as vital links, guiding researchers to relevant materials through academic databases, and embodied the core ideas of a research topic; their absence makes finding pertinent documents challenging. Author keywords, self-selected by authors, represent the essence of their work. In this review, Fig. 2 shows the top 10 author keywords in the dataset. "machine learning" led with 212 instances, reflecting its prominence in XAI and PdM research. Following were "explainable artificial intelligence" and "deep learning", with 208 and 145 occurrences, respectively. These author-provided keywords efficiently navigate the knowledge landscape of scientific fields.

4) KNOWLEDGE SYNTHESIS AND CONCEPTUAL STRUCTURE

Science mapping, a pursuit to unravel the network within evolving scientific knowledge [54], aimed to uncover the structure and dynamics of scientific research. It offered a statistical lens to delve into scientific domains, highlighting key themes and developments. Using the Walktrap algorithm [55], the interactive co-occurrence grid visualised in Fig. 3, standardised associations, and linked words within a unified document; this approach illuminated topics and insights within a research field and mapped the evolutionary path of studies over time. Fig. 3's network

TABLE 3. Overview of the final dataset (2012 – 2022).

Description	Results
Timespan	2012:2022
Sources (Journals, Books, among others.)	298
Documents	759
Annual growth rate%	44.85
Average age of the document	2.16
Average citations per doc	14.24
References	35274
Document contents	
Keywords Plus	3878
Author's Keywords	2564
Authors	
Authors	3559
Authors of single-authored docs	24
Authors' Collaboration	
Single-authored docs	24
Co-Authors per Doc	5.43
International co-authorships%	20.42
Document types	
Article	759

portrays the progression of XAI in PdM research, demonstrating the frequent co-occurrence of "explainable artificial intelligence" and "deep learning", underscoring the growing emphasis on integrating explainability into DL algorithms.

5) SOCIAL STRUCTURE

Fig. 4 highlighted Wang Y and Zhang Y as prominent contributors with significant collaboration scores in the field. Interestingly, their collaborative efforts were not directed toward each other.

Fig. 5 shows that prominent institutions like the University of Florida, Tsinghua University, and Central South University held significant ranks in collaboration. However, their collaborative efforts remain separate, without intersections.

Similarly, Fig. 6 revealed that the United States of America (USA) predominantly collaborated with China and India in its research efforts. Following this detailed bibliometric analysis, the next section presents the critical analysis and identified research gaps, elucidating this study's rationale.

F. CRITICAL ANALYSIS

Analysing research gaps is crucial to highlighting what is lacking in the literature and driving this research effort. It reveals sectors that need XAI and influences the methodologies chosen. Bibliometric analysis also leads to the need for XAI. The review identified three sectors using PdM with XAI: (a) transportation, (b) manufacturing, and (c) smart grids. Among these, manufacturing garnered the attention of

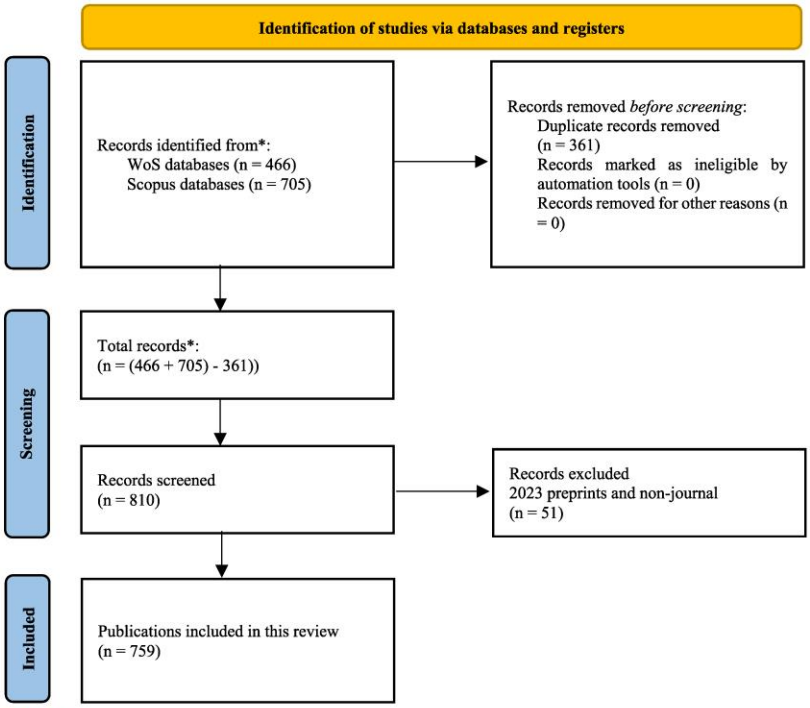

FIGURE 1. Summary of the data extraction and screening process.

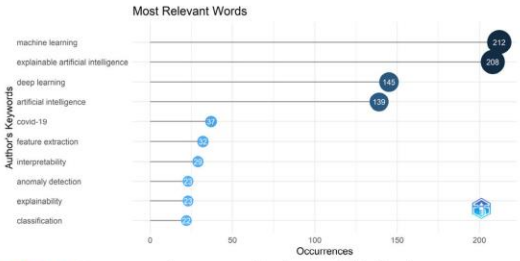

FIGURE 2. Top 10 most frequent author keywords in the dataset.

FIGURE 3. Co-occurrence network.

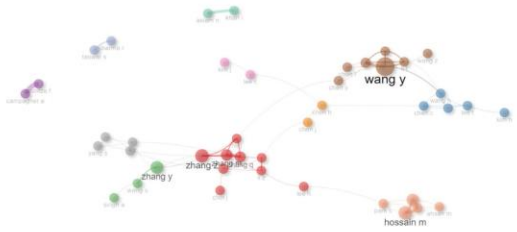

FIGURE 4. Collaboration network by author.

FIGURE 5. Collaboration network by institution.

twelve studies [31], [32], [33], [34], [35], [36], [37], [38], [40], [42], [43], [44], while transportation claimed two [30], [41], and smart grids only one [39].

Various state-of-the-art models, like Ensemble Learning (EL) [36], [44], and Deep Learning [30], [33], [37], were used in PdM; this study will use similar algorithms, aiming for comparable results and employing metrics like F1 score,

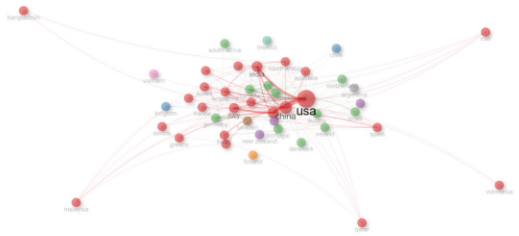

FIGURE 6. Collaboration network by country.

accuracy, and ROC-AUC for evaluation [30], [32], [33], [44], [56], [57]. Also, due to the prevalent usage of the test F1 score and for comparative analysis, it will be used to select the most outstanding DL and ML algorithms on each dataset, which will then be used for the explainability stage.

Popular explainability techniques included ELI5, LIME, SHAP, and LRP. Where ELI5 focused on feature impact, while [39] SHAP provided global explanations [30], [32], [36], [38], [39], [40], [42], [44] and LRP suits DL models [33], [41]. Although CFEs showed their application in the outcome dimension [34], [36], this study will embrace them to address the end-user dimension, offering choices for desired outcomes. Unfortunately, most studies do not address the data dimension's explainability [22], a gap that this research aimed to fill.

The bibliometric analysis revealed a prevailing trend of AI models being used in PdM, substantiating the literature review finding [31], [32], [36], [38], [39], [40], [42], [44] (refer to Fig. 2). Additionally, it underscored the prevalent use of XAI in DL, reinforcing the imperative need for explainability in DL (refer to Fig. 3). The social structure also showed an increasing demand for collaborative efforts to advance XAI in PdM research (refer to Fig. 4, 5, and 6). Notable gaps include data, outcome, end-user explainability, and the need for global explainability combined with model-specific and model-agnostic methods. There is also a gap in integrating anomaly detection and diagnostics, and a lack of focus on diagnostics alone. The intersection of XAI and PdM in SAF was under-researched, impeding farmers' understanding and usage of AI. Therefore, this study proposed a combined model that merges XAI and PdM to (1) predict maintenance needs and (2) provide explanations for the predictions made by the model, trained on a time series dataset of maintenance records, sensor readings dataset, for machine and water pump statuses. The proposed AI model predicted potential failures (prognostics), highlighting their root causes by component level (diagnostics). Also, the model addressed four dimensions: (1) data, (2) model, (3) outcome, and (4) end-user [22], [47].

III. EXPERIMENTAL DESIGN

This study aimed to predict maintenance needs and provide explanations using XAI. Also, it outlines the experimental design, covering the datasets, research design, system and parameter configurations. To achieve its objectives, the study conducted various experiments.

TABLE 4. Telemetry attributes.

No	Attribute	Description
1	datetime	When the reading was recorded
2	machineID	Unique ID for each machine
3	volt	The potential difference between two points in an electrical circuit
4	rotate	The speed at which the pump motor rotates
5	pressure	The effect at which water is carried inside the pipe
6	vibration	Rapid back-and-forth motion of the water pump
7	errorID	Five unique error types that a machine can have ['error1', 'error4', 'error3', 'error5', 'error2']
8	comp	There are four types of components for which a machine goes for maintenance ['comp2', 'comp1', 'comp4', 'comp3']
9	model	Four unique machine models ['model3', 'model4', 'model2', 'model1']
10	age	Age of machine
11	failure	Failure due to a specific component ['comp1', 'comp3', 'comp4', 'comp2']

A. DATA

Additionally, this study used open-source data to experiment with and assess the proposed AI model, enhancing reproducibility. It used the Telemetry for Predictive Maintenance dataset [58] with 11 attributes, and the Pump Sensor dataset [59] with 54 attributes. The Telemetry data covered hourly intervals from 06:00:00 on the first day of 2015 to the same time on the first day of 2016. The Pump Sensor data spanned from midnight, on 1 April 2018, to one minute before midnight on 31 August 2018, recording every minute and included data from 52 sensor units before pre-processing. The attributes and definitions within the Telemetry Predictive Maintenance dataset are shown in Table 4. The attribute and meaning of the Pump Sensor dataset are shown in Table 5.

After pre-processing, the Telemetry for Predictive Maintenance dataset had 292019 observations with 19 unique attributes, while the Pump Sensor dataset comprised 218880 observations with 4 attributes. Both datasets revealed a significant class imbalance. In the Telemetry dataset, normal (class none) instances exceeded failure instances (comp1, comp2, comp3, comp4) by 98%. Similarly, in the Pump Sensor dataset, normal status instances surpassed abnormal (recovering, broken) by 93%. Cost-Sensitive Machine

TABLE 5. Pump sensor attributes.

No	Attribute	Description
1	datetime	When the reading was recorded
2	sensor_n	Specific sensor and reading (n = 52)
3	pump_status	Three pump states ['normal', 'broken', 'recovering']

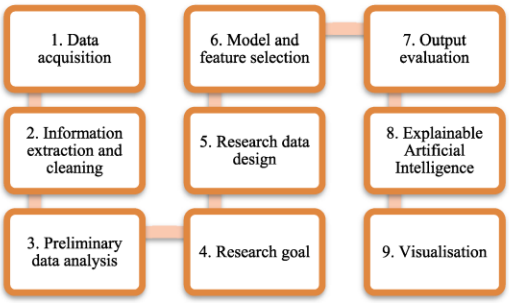

FIGURE 7. Modified SRBD stages.

Learning (CSML) was applied before the modelling stage to address these imbalances, ensuring fair representation and accuracy across different classes.

B. RESEARCH DESIGN

Influenced by the objectives of this study, the research design adopted the Systematic Research on Big Data (SRBD) methodology [60]. Despite critiques about SRBD's lack of role clarity [61], its data-driven, agile nature suited this academic research, enhancing reproducibility. SRBD typically involves seven stages: information extraction and cleaning, preliminary analysis, defining research goals, data design, model and feature selection, output evaluation, and visualisation; this study expanded to nine stages, each tailored to specific objectives and shown in Fig. 7. Stage 1, data acquisition, was pivotal for later stages. Stage 2 focused on XAI, deciphering predictions made by the model. A thorough SLR with bibliometric analysis commenced, addressing the (i) landscape of AI models and the need for explainability. Stages 2 to 6 concentrated on (ii) developing an AI model for PdM, stage 7 (iii) evaluating it, and the final stages, 8 and 9, aimed at (iv) providing explanations for the model's predictions, meeting all research objectives comprehensively.

In this study, data acquisition denoted the sourced machine and irrigation system data, analysing component-specific failures and pump failures. Information extraction and cleaning involves data pre-processing, rendering the data fit for subsequent stages [52]; this procedure encompassed (1) data cleaning, (2) data formatting, (3) feature engineering, and (4) data integration.

Additional information on these are provided as follows: (i) Data cleaning. The study employed "forward fill" for missing values, avoiding data leakage, also known as Last Observation Carried Forward (LOCF), which replaces missing values with their preceding present value, honouring the dataset's temporal order. Moreover, this method is effective when working with time series data [62], [63]. (ii) Data formatting deals with categorised variables for efficient processing. (iii) Feature engineering was used to extract features: (a) dateTime features, (b) lag features, (c) window features, and (d) periodic cyclic features. (iv) Data integration relates to merged multiple files in the Telemetry for Predictive Maintenance dataset [58]. Moreover, preliminary data analysis is focused on data analysis.

The research goal is aimed at predicting maintenance needs and providing explanations using XAI. Research data design involves structured data for AI modelling, ensuring temporal order during encoding and scaling, and split data into 70% training and 30% testing sets. In terms of the model and feature selection, the selected models and features were based on the literature review, using the test F1 score as the benchmark and for comparative analysis [32], [44]. The following were implemented in terms of the feature selection process: (1) Variance Thresholding (VT), (2) pairwise correlation, (3) Recursive Feature Elimination with Cross-Validation (RFECV), and (4) Boruta-SHAP.

For the output evaluation, the F1 score, accuracy, ROC-AUC, and class weights were used for performance evaluation, which were widely used in academic literature and those particularly tailored for dealing with imbalanced data [32], [33], [44], [56], [57]. A classifier could land within any category of the confusion matrix: *TP, TN, FP, or FN*. Where: TP denotes True Positive (correctly identified), TN denotes True Negative (correctly rejected), FP denotes False Positive (incorrectly identified), and FN denotes False Negative (incorrectly rejected). Moreover, the performance evaluation of a classification algorithm hinges on using a confusion matrix; this matrix presents correct classifications compared with erroneous ones, all categorised per class [64], [65]. Furthermore, the F1 score embodies the harmonic average, combining recall and precision; the former gauges the accurate prediction of true positives, whilst the latter quantifies the misrepresentation of the positive class. A mathematical definition of the F1 score metric emerges in Equation (3). Note that the precision and recall must first be computed:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{3}$$

In this context, the accuracy shows the proportion of accurate classifications relative to the total classifications. A mathematical definition of the accuracy metric emerges

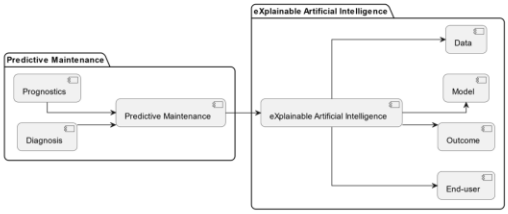

FIGURE 8. PdM and XAI illustration.

in Equation (4):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

The ROC visually shows model effectiveness, while the AUC numerically grades it from 0 to 1 [66], [67]. The AUC's versatility in appraising performance across various thresholds renders it crucial, mainly when classes differ in significance or risk. This study used the Trapezoidal rule for AUC calculation, among other methods like Riemann Sums and Simpson's rule; this approach is particularly relevant here, as accurate failure prediction is deemed more critical than correctly identifying non-failures, reflecting the unequal importance of different classes in this context. More so, the design involving eXplainable Artificial Intelligence is targeted toward transparency in predictions across four dimensions: (1) data, (2) model, (3) outcome, and (4) end-user [22], [47].

Summarily, this research advanced a unified model that fused XAI with PdM to (1) predict maintenance needs and (2) furnish explanations for the model's predictions, as seen in Fig. 8. Moreover, trained on a time series dataset containing maintenance logs, sensor data, and machine and water pump statuses, the proposed AI model predicts potential failures (prognostics), elucidating their underlying causes by component level (diagnostics) and addressing explainability through four dimensions: (1) data, (2) model, (3) outcome, and (4) the end-user [22], [47]. Finally, the study explored the option of using SHAP plots and CFE tables for intuitive explanation and visualisation presentation.

C. SYSTEM AND PARAMETER CONFIGURATION

Research experiments were done using an Intel(R) Core (TM) i5.10500H Central Processing Unit (CPU) at 2.50 GHz speed, and 16174 MB RAM, running on Windows 11 Pro 64-bit Operating System (OS) with a 6009 MB Nvidia GeForce RTX 3060 laptop Graphics Processing Unit (GPU). The Python 3.8.17 language underpinned the programming. The research detailed configurations for top-performing DL and ML algorithms per dataset. It also outlined parameter settings for the four explainability dimensions: (1) data, (2) model, (3) outcome, and (4) end-user [22], [47]. The sequence began with DL configurations for the Telemetry for Predictive Maintenance dataset, followed by those for explainability.

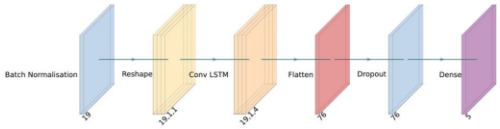

FIGURE 9. DL - Convolutional long-short-term memory neural network parameters (telemetry for predictive maintenance).

It then presented DL configurations for the Pump Sensor dataset and their explainability aspects, with a similar pattern for ML configurations.

1) TELEMETRY FOR PREDICTIVE MAINTENANCE DATASET CONFIGURATIONS

This study divided its dataset, allocating 70% to training and 30% to testing. A consistent random state, fixed at 777, was applied throughout all experiments. The model input comprised 19 features, with output targeting classification across 5 classes.

a: DEEP LEARNING (TELEMETRY FOR PREDICTIVE MAINTENANCE)

The Convolutional Long-Short-Term Memory (LSTM) Neural Network's parameters, displayed in Fig. 9, involve a sequence where input data are first transformed, normalised, and scaled; this ensures efficient training. The data are then reshaped into a (19, 1, 1) configuration, followed by a 1D Convolutional LSTM with four output channels. Subsequently, the input is flattened into a vector, retaining its initial form. To enhance regularisation, a certain percentage of inputs are randomly dropped. The final step involves a fully connected layer with 5 output units, facilitating the classification into 5 classes.

i) DATA EXPLAINABILITY (TELEMETRY FOR PREDICTIVE MAINTENANCE)

Table 6 shows the parameters for the Deepchecks data explanations.

TABLE 6. Deepchecks data explainability parameters (telemetry for predictive maintenance).

Parameter	Value	Description
n_samples	len(dataset)//2	The number of samples to use in the data integrity check
timeout	600	The maximum amount of time (in seconds) to allow for the check
n_top_columns	50	The number of top columns to include in the check
n_to_show	50	The number of results to show

ii) MODEL EXPLAINABILITY (TELEMETRY FOR PREDICTIVE MAINTENANCE)

Table 7 shows the parameters for the DL SHAP model explanations.

TABLE 7. DL - SHAP model explainability parameters (telemetry for predictive maintenance).

Parameter	Value	Description
sample_size	25	Number of samples taken from the training data for SHAP analysis
X_sub	df_train	Subset of the training data (excluding the target variable) used for SHAP analysis
explainer	SHAP explainer object	An object that can calculate SHAP values, created using the model's prediction function and sampled data
shap_values_multiple	df_test[0:sample_size]	The SHAP values for the test data
class_indices	All five classes	List of class indices for which SHAP summary plots will be generated
class_label	['comp1', 'comp2', 'comp3', 'comp4', 'none']	The label of the class corresponding to the current class index

iii) OUTCOME EXPLAINABILITY (TELEMETRY FOR PREDICTIVE MAINTENANCE)

Table 8 shows the parameters for the explanations of the DL SHAP outcomes.

TABLE 8. DL - SHAP Outcome explainability parameters (telemetry for predictive maintenance).

Parameter	Value	Description
sample_size	25	Number of samples taken from the training data for SHAP analysis
X_sub	df_train	Subset of the training data (excluding the target variable) used for SHAP analysis
explainer	SHAP explainer object	An object that can calculate SHAP values, created using the model's prediction function and sampled data
shap_values	df_test[0:1]	The SHAP values for the test data
class_indices	All five classes	List of class indices for which SHAP summary plots will be generated
class_label	['comp1', 'comp2', 'comp3', 'comp4', 'none']	The label of the class corresponding to the current class index

iv) END-USER EXPLAINABILITY (TELEMETRY FOR PREDICTIVE MAINTENANCE)

Table 9 shows the parameters for the DL DiCE end-user explanations.

TABLE 9. DL - DiCE end-user explainability parameters (telemetry for predictive maintenance).

Parameter	Value	Description
data_object	dice_ml.Data object	DiCE data object prepared using the input dataset
backend	'TF'+tf.__version__[0]	TensorFlow backend version used (e.g., 'TF2' for TensorFlow version 2.x)
model_object	dice_ml.Model object	DiCE model object created using the TensorFlow model
explainer	dice_ml.Dice object	DiCE explanation object instantiated with the data object, model object, and the method set to 'random'
desired_classes	All other classes	List of desired classes for which counterfactuals are generated
test_query	DataFrame object	Subset of a dataset (excluding the target variable) representing a test query for which counterfactuals are generated
total_CFs	5	Total number of counterfactual instances to generate for each desired class
features_to_vary	"all"	Specification of features to vary during counterfactual generation
proximity_weight	1.5	Weight is assigned to proximity in the counterfactual generation process. Feature-wise distance from the original input
diversity_weight	1.0	The weight assigned to diversity in the counterfactual generation process. Feature-wise distance between each counterfactual pair
stopping_threshold	0.5	Threshold for stopping the counterfactual generation process

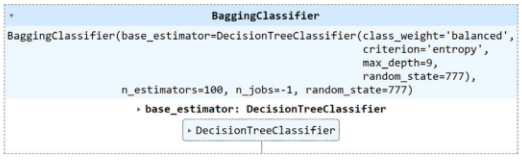

FIGURE 10. ML - Bagging classifier parameters (telemetry for predictive maintenance).

b: MACHINE LEARNING (TELEMETRY FOR PREDICTIVE MAINTENANCE)

Bagging uses an ensemble of Decision Trees (DT) (Fig. 10). Weights find themselves assigned to classes, harmonising model performance. The entropy criterion comes into play for DT node splitting. The maximum depth of the DT receives specification. The ensemble boasts 100 DT estimators. Processor cores maximised (-1) for parallel processing, accelerating model training. A random state, fixed at 777, persists, applying to all experimental procedures.

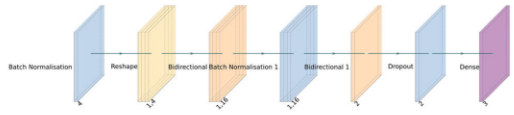

FIGURE 11. DL - Bidirectional recurrent neural network and Long-Short-Term memory neural network parameters (pump sensor).

TABLE 10. Deepchecks data explainability parameters parameters (pump sensor).

Parameter	Value	Description
n_samples	len(dataset)//2	The number of samples to use in the data integrity check
timeout	600	The maximum amount of time (in seconds) to allow for the check
n_top_columns	50	The number of top columns to include in the check
n_to_show	50	The number of results to show

2) PUMP SENSOR DATASET CONFIGURATIONS

The dataset was split, dedicating 70% for training and the rest for testing, ensuring uniformity in all trials with a random state of 777. It involved 4 distinct features as inputs and 3 classes as output for classification.

a: DEEP LEARNING (PUMP SENSOR)

In Fig. 11, the Bidirectional Recurrent Neural Network (Bi-RNN) and Long-Short-Term Memory Neural (LSTM) Network parameters were meticulously set. The input data are first transformed, normalised, and scaled. The data are then reshaped into a (1, 4) configuration, followed by a Bi-RNN and LSTM with 16 output channels. To enhance regularisation, a certain percentage of inputs are randomly dropped. The final step involves a dense layer with 3 output units, facilitating the classification into 3 classes.

i) DATA EXPLAINABILITY (PUMP SENSOR)

Table 10 shows the parameters for the Deepchecks data explanations.

ii) MODEL EXPLAINABILITY (PUMP SENSOR)

Table 11 shows the parameters for the DL SHAP model explanations.

iii) OUTCOME EXPLAINABILITY (PUMP SENSOR)

Table 12 shows the parameters for the explanations of the DL SHAP outcomes.

iv) END-USER EXPLAINABILITY (PUMP SENSOR)

Table 13 shows the parameters for the DL DiCE end-user explanations.

b: MACHINE LEARNING (PUMP SENSOR)

The parameter configurations for the Adaptive Boosting classifier are shown in (Fig. 12). The learning rate is set to 0.2, dictating the step size for model updates. The Adaptive Boosting classifier includes 100 estimators (weak models) by definition. A random state, fixed at 777, persists, applying to all experimental procedures.

TABLE 11. DL - SHAP Model explainability parameters (pump sensor).

Parameter	Value	Description
sample_size	25	Number of samples taken from the training data for SHAP analysis
X_sub	df_train	Subset of the training data (excluding the target variable) used for SHAP analysis
explainer	SHAP explainer object	An object that can calculate SHAP values, created using the model's prediction function and sampled data
shap_values_multiple	df_test[0:sample_size]	The SHAP values for the test data
class_indices	All five classes	List of class indices for which SHAP summary plots will be generated
class_label	['comp1', 'comp2', 'comp3', 'comp4', 'none']	The label of the class corresponding to the current class index

TABLE 12. DL - SHAP Outcome explainability parameters (pump sensor).

Parameter	Value	Description
sample_size	25	Number of samples taken from the training data for SHAP analysis
X_sub	df_train	Subset of the training data (excluding the target variable) used for SHAP analysis
explainer	SHAP explainer object	An object that can calculate SHAP values, created using the model's prediction function and sampled data
shap_values	df_test[0:1]	The SHAP values for the test data
class_indices	All five classes	List of class indices for which SHAP summary plots will be generated
class_label	['comp1', 'comp2', 'comp3', 'comp4', 'none']	The label of the class corresponding to the current class index

```
AdaBoostClassifier
AdaBoostClassifier(learning_rate=0.2, n_estimators=100, random_state=777)
```

FIGURE 12. ML - Adaptive boosting classifier parameters (pump sensor).

IV. RESULTS AND DISCUSSION

This section presents the results of XAI and PdM in SAF in distinct sub-themes. The research delved into prognostics and diagnostics for PdM. Prognostics predicts water pump conditions, whereas diagnostics identifies specific machine failure components. Both use DL and ML algorithms. The study also explored the "data dimension", assessing how data limitations and expectations influence insight extraction. Moreover, this section presents global explainability, aiming to unravel

TABLE 13. DL - DiCE end-user explainability parameters (pump sensor).

Parameter	Value	Description
data_object	dice_ml.Data object	DiCE data object prepared using the input dataset
backend	'TF'+tf.__version__[0]	TensorFlow backend version used (e.g., 'TF2' for TensorFlow version 2.x)
model_object	dice_ml.Model object	DiCE model object created using the TensorFlow model
explainer	dice_ml.Dice object	DiCE explanation object instantiated with the data object, model object, and the method set to 'random'
desired_classes	All other classes	List of desired classes for which counterfactuals are generated
test_query	DataFrame object	Subset of a dataset (excluding the target variable) representing a test query for which counterfactuals are generated
total_CFs	5	Total number of counterfactual instances to generate for each desired class
features_to_vary	"all"	Specification of features to vary during counterfactual generation
proximity_weight	1.5	Weight is assigned to proximity in the counterfactual generation process. Feature-wise distance from the original input
diversity_weight	1.0	The weight assigned to diversity in the counterfactual generation process. Feature-wise distance between each counterfactual pair
stopping_threshold	0.5	Threshold for stopping the counterfactual generation process

the model's behaviour through model-centric approaches for the "model dimension". Local explainability strives to explain single prediction instances, providing insights for the "outcome dimension". Addressing the "end-user dimension" emphasises creating explanations that balance abstraction and detail. In addition, discussions detail a comparative analysis with related studies, while following themes (prognostics, diagnostics, data, model, outcome, and end-user) and results interpretation, exposing implications and alignment to the related literature; this section concludes by summarising results and discussion.

Recall, that this research had a dual aim to: (1) predict maintenance needs and (2) provide explanations using XAI in PdM for SAF. Section II conducted an SLR with bibliometric analysis, tackling the first research objective: (a) conducting an SLR using a bibliometric analysis to determine the current landscape of AI models and the need for XAI. Section III addressed the following three objectives of this research to achieve its aim: (b) developing an AI model to predict maintenance needs, (c) evaluating the proposed AI model, (d) then identifying and providing explanations for the predictions made by the proposed AI model.

A. EXPERIMENTAL RESULTS

The experimental results reveal insights from the PdM alongside the XAI experiments. First, it dissects the results of both the employed DL and ML models. Next, it shifts towards XAI, casting light on the four explainability dimensions drawn from scholarly literature: (1) data, (2) model, (3) outcome, and (4) end-user.

1) PREDICTIVE MAINTENANCE RESULTS

In Section II, the research navigated prognostics and diagnostics for PdM in SAF, focusing on system failure and component failure prediction literature. The Telemetry for Predictive Maintenance dataset, initially with 11 attributes, expanded to 40 through feature engineering and refined to 19 for optimised performance, dismissing non-essential attributes like "Datatime", "MachineID", and "model". Similarly, the Pump Sensor dataset, starting with 54 attributes, was distilled to 214, then to 4 after releasing irrelevant or incomplete attributes. With high test F1 scores (above 92%), accuracies (above 90%), and ROC-AUCs (above 80%), the models exhibited strong adaptability to new data, avoiding overfitting. DT and CatBoost classifiers, while slightly trailing, still delivered competitive results. These results, detailed in Tables 14 to 17, underscore the proposed model's efficacy as a leading solution for PdM in SAF.

a: DEEP LEARNING (PREDICTIVE MAINTENANCE RESULTS)

Table 14 shows the DL classifier comparison on the Telemetry for Predictive Maintenance dataset.

TABLE 14. DL - classifier comparison (telemetry for predictive maintenance).

Classifier	Train F1 score	Test F1 score	Overfitting	Train accuracy	Test accuracy	Test ROC-AUC
Convolutional LSTM Neural Net	0.958015	**0.970003**	False	0.942831	**0.942831**	0.967616
BiRNN LSTM Neural Net	0.95516	0.963522	False	0.945424	0.945424	0.957594
Artificial Neural Net	0.945871	0.95583	False	0.922676	0.922676	0.939171
Convolutional Neural Net	0.936568	0.939713	False	0.905862	0.905862	**0.908591**

Table 15 shows the DL classifier comparison on the Pump Sensor dataset.

TABLE 15. DL - classifier comparison (pump sensor).

Classifier	Train F1 score	Test F1 score	Overfitting	Train accuracy	Test accuracy	Test ROC-AUC
BiRNN LSTM Neural Net	0.862548	**0.998576**	False	0.90576	0.998782	0.928758
Artificial Neural Net	0.861264	0.998264	False	0.905963	**0.998843**	0.802616
Convolutional Neural Net	0.948287	0.997471	False	0.924936	0.996573	0.800625
Convolutional LSTM Neural Net	0.955991	0.960525	False	0.933075	0.92518	**0.935467**

b: MACHINE LEARNING (PREDICTIVE MAINTENANCE RESULTS)

Table 16 shows the ML classifier comparison on the Telemetry for Predictive Maintenance dataset.

TABLE 16. ML - classifier comparison (telemetry for predictive maintenance).

Classifier	Train F1 score	Test F1 score	Overfitting	Train accuracy	Test accuracy	Test ROC-AUC
Bagging classifier	0.993066	**0.994729**	False	0.992466	**0.992466**	0.994373
LGBM classifier	0.993131	0.99447	False	0.992275	0.992275	0.993802
Decision Tree classifier	0.993036	0.994258	False	0.992432	0.992432	0.993836
XGB classifier	0.993043	0.993304	False	0.993733	0.993733	0.993779
RandomForest classifier	0.985309	0.985562	False	0.982755	0.982755	0.982501
CatBoost classifier	0.983746	0.984394	False	0.98028	0.98028	**0.980435**
AdaBoost classifier	0.969947	0.97304	False	0.977849	0.977849	0.981074

Table 17 shows the ML classifier comparison on the Pump Sensor dataset.

Furthermore, as established earlier, applying the test F1 score calls for its use in selecting distinguished DL and ML algorithms per dataset [32], [44]. Thus, the Convolutional LSTM Neural Network, BiRNN LSTM Neural Network, Bagging classifier, and AdaBoost classifier algorithms were used for the explainability stage.

TABLE 17. ML - classifier comparison (pump sensor).

Classifier	Train F1 score	Test F1 score	Overfitting	Train accuracy	Test accuracy	Test ROC-AUC
AdaBoost classifier	0.945365	**0.999525**	False	0.951154	**0.999574**	**0.969477**
RandomForest classifier	0.99794	0.998902	False	0.997931	0.998766	0.936051
LGBM classifier	0.996472	0.998612	False	0.996417	0.998919	0.905535
XGB classifier	0.995269	0.998347	False	0.995255	0.998051	0.906245
Decision Tree classifier	0.971186	0.992829	False	0.969703	0.98803	0.494588
Bagging classifier	0.971175	0.992821	False	0.96969	0.988015	0.832808
CatBoost classifier	0.886011	0.951349	False	0.838352	0.90922	0.492795

2) EXPLAINABLE ARTIFICIAL INTELLIGENCE RESULTS

Section III's "Data" segment, showed two datasets: Telemetry for Predictive Maintenance and Pump Sensor, each with 5 and 3 classes, respectively. The analysis considers these class structures, unless otherwise indicated. Key variables like "age", "datetime_day_of_month", "datetime_hour_sin", and "datetime_week" transformed into float values post-standardisation; this section presents and discusses explainability dimensions, using DL algorithms as input tools.

a: INSIGHTS FROM THE DATA

Using data-centric methods, the study extracted deep insights directly from the data, assessing its purity and quality. Results showed meticulous dataset preparation, adhering to consistency, integrity, and predictability criteria. For instance, the Telemetry for Predictive Maintenance dataset featured 21 columns, surpassing the original 19. In contrast, the Pump Sensor dataset had 6, exceeding the original 4. However, these extra columns, included categorical features and target variables, which were essential metadata, influencing neither data purity checks nor overall test outcomes. Data evaluations in Table 18 showcased successful checks, confirming diverse values, the absence of excessive special characters, uniform handling of null values, minimal variation in data types, string content uniformity, lack of duplicate data, and relevant correlations under the threshold. A minor discrepancy emerged, with 0.02% of the samples showing conflicting labels, narrowly exceeding the set limit. Similarly, Table 19 affirmed diversity in columns, appropriate handling of special characters and null values, minimal data-type variation, string uniformity, and no duplicate data. Although, feature-feature correlations and predictive power of features like "sensor_05", with a score of 0.93, stood out, a single test narrowly missed the mark. These evaluations showed the

TABLE 18. Data purity (telemetry for predictive maintenance).

Status	Check	Condition	More Information
Passed	Single Value in Column	It does not contain only a single value	Passed for 21 relevant columns
Passed	Special Characters	The ratio of samples containing special characters is less or equal to 0.1%	Passed for 21 relevant columns
Passed	Mixed Nulls	The number of different null types is less or equal to 1	Passed for 21 relevant columns
Passed	Mixed Data Types	Rare data types in column are either more than 10% or less than 1% of the data	21 columns passed: found 0 columns with negligible types mix, and 21 columns without any types mix
Passed	String Mismatch	No string variants	Passed for one relevant column
Passed	Data Duplicates	Duplicate data ratio is less or equal to 5%	Found 0% duplicate data
Passed	String Length Out of Bounds	The ratio of string length outliers is less or equal to 0%	No relevant columns to check were found
Passed	Feature-Label Correlation	Features' Predictive Power Score (PPS) is less than 0.8	Passed for 19 relevant columns
Passed	Feature-Feature Correlation	Not more than 0 pairs are correlated above 0.9	All correlations are less than 0.9 except for pairs
Failed	Conflicting Labels	Ambiguous sample ratio is less or equal to 0%	The ratio of samples with conflicting labels: 0.02%

TABLE 19. Data purity (pump sensor).

Status	Check	Condition	More Information
Passed	Single Value in Column	It does not contain only a single value	Passed for six relevant columns
Passed	Special Characters	The ratio of samples containing special characters is less or equal to 0.1%	Passed for six relevant columns
Passed	Mixed Nulls	The number of different null types is less or equal to 1	Passed for six relevant columns
Passed	Mixed Data Types	Rare data types in column are either more than 10% or less than 1% of the data	Six columns passed: found 0 columns with negligible types mix, and six columns without any types mix
Passed	String Mismatch	No string variants	Passed for one relevant column
Passed	Data Duplicates	Duplicate data ratio is less or equal to 5%	Found 0% duplicate data
Passed	String Length Out of Bounds	The ratio of string length outliers is less or equal to 0%	No relevant columns to check were found
Passed	Conflicting Labels	Ambiguous sample ratio is less or equal to 0%	The ratio of samples with conflicting labels: 0%
Passed	Feature-Feature Correlation	Not more than 0 pairs are correlated above 0.9	All correlations are less than 0.9 except for pairs
Failed	Feature-Label Correlation	Features' Predictive Power Score is less than 0.8	Found 3 out of 4 features with PPS above threshold: sensor_05: 0.93 sensor_10_window_3H_mean: 0.83 sensor_12_window_3H_mean: 0.81

passing of 9 out of 10 tests in both datasets, underscoring the datasets' robustness and reliability, setting a strong foundation for subsequent PdM and AI analysis.

i) DATA RESULTS

Table 18 shows the data purity tests on the Telemetry for Predictive Maintenance dataset. Table 19 shows the data purity tests on the Pump Sensor dataset.

b: INSIGHTS FROM THE MODEL

A focus on global explainability emerged, and delved into model-centric methodologies that explore links between input features and outcomes; this approach yielded rich insights directly from the model, primarily through SHAP plots, which clarify feature impact direction using colour-coded bars (blue for negative, red for positive). For clarity, plot titles like "class 'compX'" indicate "compY's" influence on the target "compX", underpinning precise component-level PdM. Unlike feature importance, SHAP values provide a nuanced view of feature attribution, applying game theory for a comprehensive impact assessment [68], [69]. Also, unlike feature importance, which ranks features by performance but lacks detailed impact analysis [70], SHAP values offer a balanced view, suitable for both global and local effects. The research also leverages SHAP for global feature attribution using DL algorithms, providing a holistic understanding of model behaviour. For instance, Fig. 13 shows how the "error1count_1" feature significantly influences predictions for class "comp1," with a notable SHAP value of 0.09. Similarly, Fig. 14 highlights sensor_05 as a key predictor for a "broken" pump status, shown by its leading SHAP value; this analysis, extending beyond individual predictions, helps identify critical features for effective PdM, guiding future data strategies.

i) DEEP LEARNING (MODEL RESULTS)

Fig. 13 shows the DL Global SHAP for class comp1 on the Telemetry for Predictive Maintenance dataset.

Fig. 14 shows the DL Global SHAP for class broken on the Pump Sensor dataset.

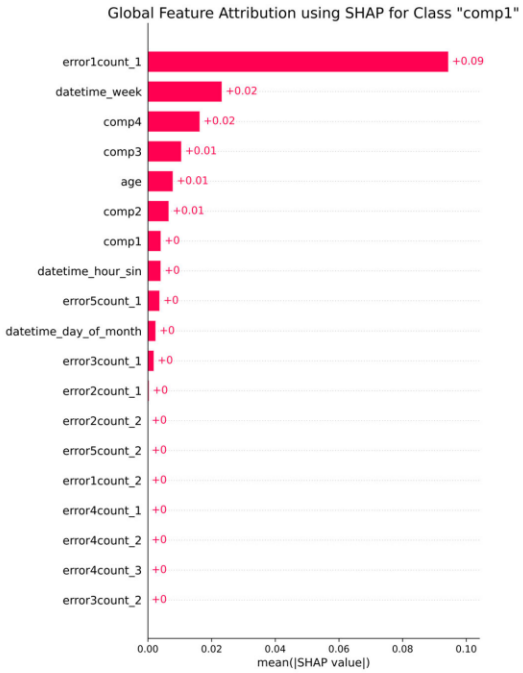

FIGURE 13. DL - Global SHAP for class comp1 (telemetry for predictive maintenance).

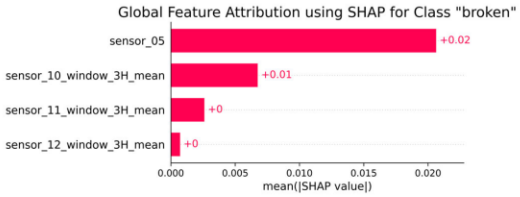

FIGURE 14. DL - Global SHAP for class broken (pump sensor).

c: INSIGHTS FROM THE OUTCOME

Local explainability methods illuminate individual prediction instances, extending the model's insights. SHAP, central to this analysis, analyses single-instance feature impact. Features like "error1count_2" quantify specific error occurrences, while "datetime_hour_sin" and "datetime_hour_cos" emerge from sine and cosine transformations (detailed in Section III's "Periodic Cyclic Features"). The study then applied SHAP to DL for local feature attribution in the outcome dimension, providing detailed insights. Unlike global attribution, local SHAP provides granular explainability. For instance, in Fig. 15, "error1count_1" significantly sways the prediction towards "comp1". Other features like "comp4" and "comp2" also impact this prediction, albeit less so. On the contrary,

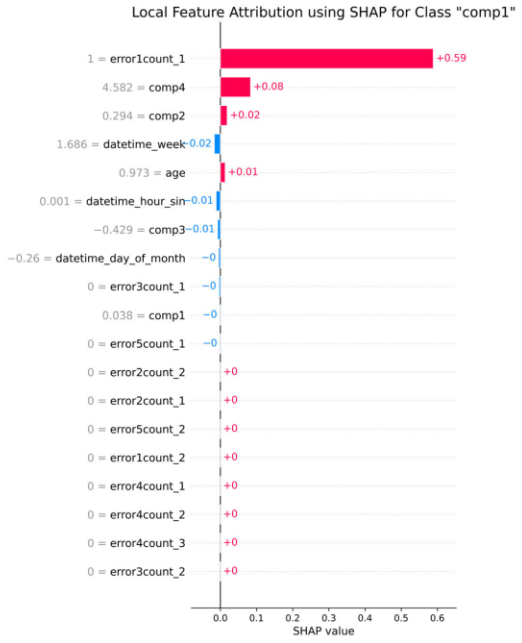

FIGURE 15. DL - Local SHAP for class comp1 (telemetry for predictive maintenance).

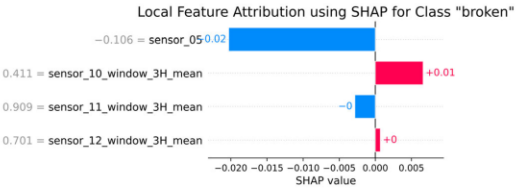

FIGURE 16. DL - Local SHAP for class broken (pump sensor).

"datetime_week", "datetime_hour_sin", and "comp3" reduce the impact of classifying as "comp1". Features such as "error3count_1" and "comp1" show zero impact on classification, highlighting the nuanced influence of different features. Fig. 16 further shows local attribution, where "sensor_05" negatively affects the "broken" pump status prediction, while "sensor_10_window_3H_mean" enhances it. However, features like "sensor_11_window_3H_mean" and "sensor_12_window_3H_mean" show no effect on the prediction; this approach underscores the specific influence of each feature on the model's decision-making process.

i) DEEP LEARNING (OUTCOME RESULTS)

Fig. 15 shows the DL Local SHAP for class comp1 on the Telemetry for Predictive Maintenance dataset.

Fig. 16 shows the DL Local SHAP for class broken on the Pump Sensor dataset.

d: INSIGHTS TOWARD THE END-USER

CFEs balance abstraction and detail. "Counterfactual n for class 'compX'" indicates altering "compY" to trigger a "compX" a broken status, aiding in pinpointing the malfunctioning component. Phrases like "The feature compY must transition" indicate necessary changes in "compY" to affect "compX". Some CFEs, without percentage change, relate to categorical features, while others, exceeding 100%, indicate the extent of change needed for a desired outcome. Light grey and light blue in the CFE tables represent decreasing and increasing feature changes, respectively. Table 20 presents "what-if" (counterfactuals) scenarios for the Telemetry for Predictive Maintenance dataset, showing changes needed to switch from class "none" to "comp1". These counterfactuals highlight the impact of features like "comp4" and "comp2", with substantial percentage changes (increases of 593.70% and decreases of 165.64%, respectively). The "errorXcount_Y" features, moving from 0.0 to 1.0 demonstrate the influence of specific errors. Counterfactuals reveal that minor adjustments, like a 22.33% change in "datetime_week", can be significant. Consistent settings of "error1count_2" and "error2count_1" to 1.0 in Counterfactual five highlight their ongoing effect on outcomes. In the Pump Sensor dataset, counterfactuals in Table 21 show shifts from "normal" to "broken". A dramatic change in "sensor_05" is required, indicating its crucial role in predicting breakdowns. Counterfactuals two and three also demand significant adjustments in "sensor_12_window_3H_mean" and "sensor_05". "Sensor_05" is consistently altered across scenarios, underscoring its impact on classifying the "broken" status.

i) DEEP LEARNING (END-USER RESULTS)

Table 20 shows the DL CFE from none to comp1 on the Telemetry for Predictive Maintenance dataset.

TABLE 20. DL - CFE from none to comp1 (telemetry for predictive maintenance).

Counterfactual one for class "comp1"
The feature comp4 must transition from -0.369327187538147 to 1.82338613 **(593.70%)**
The feature error4count_2 must transition from 0.0 to 1.0
Counterfactual two for class "comp1"
The feature comp1 must transition from 2.1010522842407227 to 4.61362886 **(119.59%)**
The feature error5count_2 must transition from 0.0 to 1.0
Counterfactual three for class "comp1"
The feature datetime_week must transition from -1.7300148010253906 to -1.3437247 **(22.33%)**
The feature error5count_2 must transition from 0.0 to 1.0
Counterfactual four for class "comp1"
The feature comp2 must transition from 1.2788968086242676 to -0.83951104 **(-165.64%)**
The feature error1count_1 must transition from 0.0 to 1.0
Counterfactual five for class "comp1"
The feature error1count_2 must transition from 0.0 to 1.0
The feature error2count_1 must transition from 0.0 to 1.0

Table 21 shows the DL CFE from normal to broken on the Pump Sensor dataset.

TABLE 21. DL - CFE from normal to broken (pump sensor).

Counterfactual one for class "broken"
The feature sensor_05 must transition from 0.1950564682483673 to -3.5593897 **(-1924.80%)**
Counterfactual two for class "broken"
The feature sensor_12_window_3H_mean must transition from 0.7513946890830994 to -2.0743027 **(-376.06%)**
The feature sensor_05 must transition from 0.1950564682483673 to -3.0883916 **(-1683.33%)**
Counterfactual three for class "broken"
The feature sensor_12_window_3H_mean must transition from 0.7513946890830994 to -2.2550662 **(-400.12%)**
The feature sensor_05 must transition from 0.1950564682483673 to -3.3266737 **(-1805.49%)**
Counterfactual four for class "broken"
The feature sensor_05 must transition from 0.1950564682483673 to -3.639838 **(-1966.04%)**
Counterfactual five for class "broken"
The feature sensor_05 must transition from 0.1950564682483673 to -3.523411 **(-1906.35%)**

B. DISCUSSION

After presenting the experimental results, the results are then discussed in the context of this research. It starts with a comparative analysis with related studies while discussing the results in a similar order as was done in the presentation of results: PdM and the XAI results. The first (PdM) discusses the results of both the DL and ML models. The second (XAI) discusses the results of XAI, focusing on the four dimensions of explainability extracted from the literature: (1) data, (2) model, (3) outcome, and (4) end-user.

1) COMPARATIVE ANALYSIS WITH RELATED STUDIES

The comparative analysis in this section offers insights into this research's unique contributions, contrasting with previous studies to deepen understanding. Performance comparisons in Table 22 demonstrate the superior functioning of DL and ML classifiers on two datasets. These models, Convolutional LSTM and BiRNN LSTM Neural Nets, excel across metrics without overfitting, highlighting their reliable generalisation. Comparisons with [33] show that this research's LSTM classifiers consistently outperformed others, with a 5.81% test accuracy improvement, calculated as an average percentage difference. Moreover, the XGBoost classifiers perform exceptionally well on diverse datasets (Table 23). For instance, in the Telemetry for Predictive Maintenance dataset, XGBoost achieves consistent F1 scores and accuracy. Compared to [44], this research's XGBoost classifier shows enhanced performance, with an increase of 7.09% in test F1 score, 10.66% in accuracy, and 4.29% in ROC-AUC;

this research's XGBoost classifier surpasses [44] in all test metrics, affirming the superiority of this approach in PdM, machine status, and water pump data analysis. These results emphasise LSTMs and Boosters' efficacy in PdM for SAF.

Table 22 shows the DL classifier comparison with related studies.

TABLE 22. DL - classifier comparison with related studies.

Source	Dataset	Classifier	Train F1 score	Test F1 score	Overfitting	Train accuracy	Test accuracy	Test ROC-AUC
This research	Telemetry for Predictive Maintenance	Convolutional LSTM Neural Net	0.958015	0.970003	False	0.942831	0.942831	**0.967616**
This research	Telemetry for Predictive Maintenance	BiRNN LSTM Neural Net	0.95516	0.963522	False	0.945424	0.945424	0.957594
This research	Pump Sensor	BiRNN LSTM Neural Net	0.862548	**0.998576**	False	0.90576	**0.998782**	0.928758
This research	Pump Sensor	Convolutional LSTM Neural Net	0.955991	0.960525	False	0.933075	0.92518	0.935467
[33]	Time series	RNN LSTM	-	-	-	-	0.9007	-

Table 23 shows the ML classifier comparison with related studies.

TABLE 23. ML - classifier comparison with related studies.

Source	Dataset	Classifier	Train F1 score	Test F1 score	Overfitting	Train accuracy	Test accuracy	Test ROC-AUC
This research	Telemetry for Predictive Maintenance	XGB classifier	0.993043	0.993304	False	0.993733	0.993733	**0.993779**
This research	Pump Sensor	XGB classifier	0.995269	**0.998347**	False	0.995255	**0.998051**	0.906245
[44]	Time series	XGB classifier	-	0.93	-	-	0.9	0.911

2) EXPLAINABILITY DIMENSIONS

a: DATA DIMENSION

Contrary to related studies, this research addresses an unexplored dimension: data explainability. Pioneering in this dimension, the study sets a precedent for future research. It methodically demonstrates how to discern data limitations and align researcher expectations. Tables 18 and 19 exhibit rigorous dataset analysis, highlighting diversity, uniformity, and minimal duplications. Despite minor label discrepancies, these datasets exemplify consistency and integrity, crucial for reliable PdM and AI analysis.

b: MODEL AND OUTCOME DIMENSIONS

This research diverges from related studies in its model and outcome explainability approach. A key premise often assumed is the independence and non-correlation of features or attributes; this assumption is fallible, introducing potential biases in the "model and outcome dimensions." Thus, data explainability becomes a foundational element for these dimensions, guiding the assessment of inherent limitations and depth of insights derivable from the data; this research's contribution lies in establishing a framework for exploring the levels of explainability achievable with specific datasets. Employing SHAP for global feature attribution in DL, it transcends mere feature ranking, enabling a comprehensive understanding of model behaviour. In local explainability, the study goes further, analysing individual prediction instances. Local methods, through SHAP, provide an in-depth feature impact analysis on single-instance predictions.

c: END-USER DIMENSION

Concerning end-user explainability, this research stands out from existing literature, particularly in explaining predictions to non-technical users; this study significantly advances end-user explainability, showcasing how to strike this balance effectively. Using the two datasets, it presented "what-if" (counterfactuals) scenarios in Tables 20 and 21, demonstrating the necessary changes to shift predictions from one class to another preferred class (outcome). These counterfactuals elucidate the impact of specific features, underscoring the significance of consistent settings in influencing outcomes.

3) SCOPE AND LIMITATIONS OF THE RESEARCH

The scope of the study was to incorporate XAI and PdM in SAF, focusing on (1) machines and (2) irrigation systems. Moreover, the study faced constraints stemming from limited PdM data, using merely two time series datasets. PdM findings focused exclusively on DL and EL for ML models, influenced by pertinent literature. Similarly, the scope of XAI was confined to (1) data, (2) model, (3) outcome, and (4) end-user explanations, framed by established dimensions of explainability. Additionally, the bibliometric analysis faced its own set of limitations; the review methodology encapsulated data processes, descriptive statistics, keyword analysis, knowledge synthesis, and the exploration of conceptual and social structures. Searches targeted "Titles", "Abstracts", and "Subject Headings" in a period from 2012 to 2022, excluding the incomplete year 2023. In the databases, "Keywords Plus" and "Keywords" were used as Subject Headings to search for synonyms, domain-specific language, and additional known relevant words [49]; this approach extracted literature relevant to the topic of this research. In addition, only two academic databases were used: WoS and Scopus. For replication and ensuring reproducibility, the researcher's experimental code is available in their GitHub repository at: https://github.com/iammelvink.

V. CONCLUSION

This research marks a significant advancement in SAF, merging XAI with PdM. It predicts maintenance requirements and elucidates model predictions; this fusion enhances understanding and application of AI for PdM in SAF, addressing

the research aim and objectives. The study illuminates the potent synergy between XAI and PdM, fostering transparent, understandable algorithmic decisions, a vital move towards clarifying the often opaque nature of DL and ML. The research showcases the impressive performance of the LSTM and XGBoost classifiers in PdM, setting new benchmarks in SAF. These advances promise improved predictive accuracy and reliability. Exploring data dimensions reveals inherent limitations and refines expectations for data-driven insights. The research unravels the model and outcome complexities using SHAP values and providing detailed, practical explanations. It also introduces varied predictive scenarios through counterfactuals, which could improve stakeholder decision-making.

Future research directions include integrating multi-modal data to enhance AI model efficacy for PdM in SAF; developing new explainability metrics; incorporating Human-in-the-Loop (HITL) systems for collaborative decision-making and enhanced PdM model reliability; analysing the effects of XAI on human decision-making in agriculture; exploring ethical aspects of agricultural AI, focusing on Fairness, Accountability, and Transparency (FAT); and assessing the long-term impact of these technologies on productivity, sustainability, and economic factors in SAF, setting the stage for future studies that are as impactful as they are essential.

REFERENCES

[1] L. W. Bell, A. D. Moore, and J. A. Kirkegaard, "Evolution in crop–livestock integration systems that improve farm productivity and environmental performance in Australia," (in English), *Eur. J. Agronomy*, vol. 57, pp. 10–20, Jul. 2014, doi: 10.1016/j.eja.2013.04.007.

[2] J. Rana and J. Paul, "Consumer behavior and purchase intention for organic food: A review and research agenda," (in English), *J. Retailing Consum. Services*, vol. 38, pp. 157–165, Sep. 2017, doi: 10.1016/j.jretconser.2017.06.004.

[3] Y. Zhong, I. K. W. Lai, F. Guo, and H. Tang, "Research on government subsidy strategies for the development of agricultural products e-commerce," (in English), *Agriculture*, vol. 11, no. 11, p. 1152, Nov. 2021, doi: 10.3390/agriculture11111152.

[4] A. Calcante, L. Fontanini, and F. Mazzetto, "Repair and maintenance costs of 4WD tractors in northern Italy," (in English), *Trans. ASABE*, vol. 56, no. 2, pp. 355–362, 2013, doi: 10.13031/2013.42660.

[5] E. Elahi, Z. Khalid, M. Z. Tauni, H. Zhang, and X. Lirong, "Extreme weather events risk to crop-production and the adaptation of innovative management strategies to mitigate the risk: A retrospective survey of rural Punjab, Pakistan," (in English), *Technovation*, vol. 117, Sep. 2022, Art. no. 102255, doi: 10.1016/j.technovation.2021.102255.

[6] M. Yildirim, N. Z. Gebraeel, and X. A. Sun, "Integrated predictive analytics and optimization for opportunistic maintenance and operations in wind farms," (in English), *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4319–4328, Nov. 2017, doi: 10.1109/TPWRS.2017.2666722.

[7] P. Zhou and P. T. Yin, "An opportunistic condition-based maintenance strategy for offshore wind farm based on predictive analytics," (in English), *Renew. Sustain. Energy Rev.*, vol. 109, pp. 1–9, Jul. 2019, doi: 10.1016/j.rser.2019.03.049.

[8] C. Eastwood, L. Klerkx, M. Ayre, and B. Dela Rue, "Managing socio-ethical challenges in the development of smart farming: From a fragmented to a comprehensive approach for responsible research and innovation," (in English), *J. Agricult. Environ. Ethics*, vol. 32, nos. 5–6, pp. 741–768, Dec. 2019, doi: 10.1007/s10806-017-9704-5.

[9] S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, "Big data in smart farming—A review," (in English), *Agricult. Syst.*, vol. 153, pp. 69–80, May 2017, doi: 10.1016/j.agsy.2017.01.023.

[10] S. A. Z. Rahman, K. C. Mitra, and S. M. M. Islam, "Soil classification using machine learning methods and crop suggestion based on soil series," in *Proc. 21st Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2018, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/8631943/

[11] V. Panchbhaiyye and T. Ogunfunmi, "Experimental results on using deep learning to identify agricultural pests," in *Proc. IEEE Global Humanitarian Technol. Conf. (GHTC)*, Oct. 2018, pp. 1–2. [Online]. Available: https://ieeexplore.ieee.org/document/8601896/

[12] P. Shankar, N. Werner, S. Selinger, and O. Janssen, "Artificial intelligence driven crop protection optimization for sustainable agriculture," in *Proc. IEEE/ITU Int. Conf. Artif. Intell. Good (AI4G)*, Sep. 2020, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/9311082/

[13] N. Taravatrooy, M. R. Nikoo, S. Hobbi, M. Sadegh, and A. Izady, "A novel hybrid entropy-clustering approach for optimal placement of pressure sensors for leakage detection in water distribution systems under uncertainty," (in English), *Urban Water J.*, vol. 17, no. 3, pp. 185–198, Mar. 2020, doi: 10.1080/1573062x.2020.1758162.

[14] D. R. Vincent, N. Deepa, D. Elavarasan, K. Srinivasan, S. H. Chauhdary, and C. Iwendi, "Sensors driven AI-based agriculture recommendation model for assessing land suitability," *Sensors*, vol. 19, no. 17, p. 3667, Aug. 2019, doi: 10.3390/s19173667.

[15] M. Kande, A. Isaksson, R. Thottappillil, and N. Taylor, "Rotating electrical machine condition monitoring automation—A review," *Machines*, vol. 5, no. 4, p. 24, Oct. 2017, doi: 10.3390/machines5040024.

[16] K. Poppe, S. Wolfert, C. Verdouw, and A. Renwick, "A European perspective on the economics of big data," *Farm Policy J.*, vol. 12, no. 1, pp. 11–19, 2015.

[17] S. Sonka, "Big data: From hype to agricultural tool," *Farm Policy J.*, vol. 12, pp. 1–9, Jan. 2015. [Online]. Available: https://www.researchgate.net/publication/279771638_Big_Data_From_Hype_to_Agricultural_Tool

[18] A. Brauneck, L. Schmalhorst, M. M. K. Majdabadi, M. Bakhtiari, U. Völker, J. Baumbach, L. Baumbach, and G. Buchholtz, "Federated machine learning, privacy-enhancing technologies, and data protection laws in medical research: Scoping review," *J. Med. Internet Res.*, vol. 25, Mar. 2023, Art. no. e41588, doi: 10.2196/41588.

[19] T. R. Chhetri, A. Kurteva, R. J. DeLong, R. Hilscher, K. Korte, and A. Fensel, "Data protection by design tool for automated GDPR compliance verification based on semantically modeled informed consent," *Sensors*, vol. 22, no. 7, p. 2763, Apr. 2022, doi: 10.3390/s22072763.

[20] A. D. Selbst and S. Barocas, "The intuitive appeal of explainable machines," (in English), *Fordham Law Rev.*, vol. 87, no. 3, pp. 1085–1139, Dec. 2018.

[21] E. Lughofer and M. Sayed-Mouchaweh, *Predictive Maintenance in Dynamic Systems*. Cham, Switzerland: Springer, 2019, doi: 10.1007/978-3-030-05645-2.

[22] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017, *arXiv:1702.08608*.

[23] D. Orn, L. Duan, Y. Liang, H. Siy, and M. Subramaniam, "Agro-AI education: Artificial intelligence for future farmers," in *Proc. 21st Annu. Conf. Inf. Technol. Educ.*, Oct. 2020, pp. 54–57, doi: 10.1145/3368308.3415457.

[24] V. P. Harmani, B. M. Himawan, M. A. Alhadi, and A. A. S. Gunawan, "Systematic literature review: Implementation of artificial intelligence in precision agriculture," in *Proc. 5th Int. Conf. Inf. Commun. Technol. (ICOIACT)*, Aug. 2022, pp. 479–484. [Online]. Available: https://ieeexplore.ieee.org/document/9971917/

[25] U. Ehsan, P. Wintersberger, Q. V. Liao, M. Mara, M. Streit, S. Wachter, A. Riener, and M. O. Riedl, "Operationalizing human-centered perspectives in explainable AI," in *Proc. Extended Abstr. CHI Conf. Human Factors Comput. Syst.*, Yokohama, Japan, May 2021, pp. 1–6, doi: 10.1145/3411763.3441342.

[26] S. Hepenstal, L. Zhang, and B. L. William Wong, "An analysis of expertise in intelligence analysis to support the design of human-centered artificial intelligence," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2021, pp. 107–112. [Online]. Available: https://ieeexplore.ieee.org/document/9659095/

[27] M. O. Riedl, "Human-centered artificial intelligence and machine learning," *Hum. Behav. Emerg. Technol.*, vol. 1, no. 1, pp. 33–36, Jan. 2019, doi: 10.1002/hbe2.117.

[28] F. Sperrle, M. El-Assady, G. Guo, R. Borgo, D. H. Chau, A. Endert, and D. Keim, "A survey of human-centered evaluations in human-centered machine learning," (in English), *Comput. Graph. Forum*, vol. 40, no. 3, pp. 543–568, Jun. 2021, doi: 10.1111/cgf.14329.

[29] C. Molnar, G. Casalicchio, and B. Bischl, "Interpretable machine learning—A brief history, state-of-the-art and challenges," in *Proc. ECML PKDD Workshops*. Berlin, Germany: Springer, 2020, ch. 28, pp. 417–431.

[30] E. Kononov, A. Klyuev, and M. Tashkinov, "Prediction of technical state of mechanical systems based on interpretive neural network model," (in English), *Sensors*, vol. 23, no. 4, p. 1892, Feb. 2023, doi: 10.3390/s23041892.

[31] B. Ghasemkhani, O. Aktas, and D. Birant, "Balanced K-star: An explainable machine learning method for Internet-of-Things-enabled predictive maintenance in manufacturing," (in English), *Machines*, vol. 11, no. 3, p. 322, Feb. 2023, doi: 10.3390/machines11030322.

[32] M. Gashi, B. Mutlu, and S. Thalmann, "Impact of interdependencies: Multi-component system perspective toward predictive maintenance based on machine learning and XAI," (in English), *Appl. Sci.*, vol. 13, no. 5, p. 3088, Feb. 2023, doi: 10.3390/app13053088.

[33] H. Wu, A. Huang, and J. W. Sutherland, "Layer-wise relevance propagation for interpreting LSTM-RNN decisions in predictive maintenance," (in English), *Int. J. Adv. Manuf. Technol.*, vol. 118, nos. 3–4, pp. 963–978, Jan. 2022, doi: 10.1007/s00170-021-07911-9.

[34] J. Jakubowski, P. Stanisz, S. Bobek, and G. J. Nalepa, "Anomaly detection in asset degradation process using variational autoencoder and explanations," (in English), *Sensors*, vol. 22, no. 1, p. 291, Dec. 2021, doi: 10.3390/s22010291.

[35] G. Hajgató, R. Wéber, B. Szilágyi, B. Tóthpál, B. Gyires-Tóth, and C. Hős, "PredMaX: Predictive maintenance with explainable deep convolutional autoencoders," (in English), *Adv. Eng. Informat.*, vol. 54, Oct. 2022, Art. no. 101778, doi: 10.1016/j.aei.2022.101778.

[36] M. Garouani, A. Ahmad, M. Bouneffa, M. Hamlich, G. Bourguin, and A. Lewandowski, "Towards big industrial data mining through explainable automated machine learning," (in English), *Int. J. Adv. Manuf. Technol.*, vol. 120, nos. 1–2, pp. 1169–1188, May 2022, doi: 10.1007/s00170-022-08761-9.

[37] S. J. Upasane, H. Hagras, M. H. Anisi, S. Savill, I. Taylor, and K. Manousakis, "A big bang-big crunch type-2 fuzzy logic system for explainable predictive maintenance," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2021, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/9494540/

[38] M. Hermansa, M. Kozielski, M. Michalak, K. Szczyrba, L. Wróbel, and M. Sikora, "Sensor-based predictive maintenance with reduction of false alarms—A case study in heavy industry," (in English), *Sensors*, vol. 22, no. 1, p. 226, Dec. 2021, doi: 10.3390/s22010226.

[39] M. Kuzlu, U. Cali, V. Sharma, and Ö. Güler, "Gaining insight into solar photovoltaic power generation forecasting utilizing explainable artificial intelligence tools," (in English), *IEEE Access*, vol. 8, pp. 187814–187823, 2020, doi: 10.1109/ACCESS.2020.3031477.

[40] O. Serradilla, E. Zugasti, J. Ramirez de Okariz, J. Rodriguez, and U. Zurutuza, "Adaptable and explainable predictive maintenance: Semi-supervised deep learning for anomaly detection and diagnosis in press machine data," (in English), *Appl. Sci.*, vol. 11, no. 16, p. 7376, Aug. 2021, doi: 10.3390/app11167376.

[41] O. Mey and D. Neufeld, "Explainable AI algorithms for vibration data-based fault detection: Use case-adadpted methods and critical evaluation," (in English), *Sensors*, vol. 22, no. 23, p. 9037, Nov. 2022, doi: 10.3390/s22239037.

[42] H. Choi, D. Kim, J. Kim, J. Kim, and P. Kang, "Explainable anomaly detection framework for predictive maintenance in manufacturing systems," (in English), *Appl. Soft Comput.*, vol. 125, Aug. 2022, Art. no. 109147, doi: 10.1016/j.asoc.2022.109147.

[43] R. Langone, A. Cuzzocrea, and N. Skantzos, "Interpretable anomaly prediction: Predicting anomalous behavior in Industry 4.0 settings via regularized logistic regression tools," (in English), *Data Knowl. Eng.*, vol. 130, Nov. 2020, Art. no. 101850, doi: 10.1016/j.datak.2020.101850.

[44] B. Steurtewagen and D. Van den Poel, "Adding interpretability to predictive maintenance by machine learning on sensor data," (in English), *Comput. Chem. Eng.*, vol. 152, Sep. 2021, Art. no. 107381, doi: 10.1016/j.compchemeng.2021.107381.

[45] M. Reif, F. Shafait, M. Goldstein, T. Breuel, and A. Dengel, "Automatic classifier selection for non-experts," (in English), *Pattern Anal. Appl.*, vol. 17, no. 1, pp. 83–96, Feb. 2014, doi: 10.1007/s10044-012-0280-z.

[46] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," (in English), *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006, doi: 10.1016/j.neucom.2005.12.126.

[47] A. B. Arrieta, N. Díaz-Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, Jun. 2020, doi: 10.1016/j.inffus.2019.12.012.

[48] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, and F. Giannotti, "A survey of methods for explaining black box models," 2018, *arXiv:1802.01933*.

[49] E. Garfield, "KeyWords plus-ISI's breakthrough retrieval method. 1. Expanding your searching power on current-contents on diskette," *Curr Contents*, vol. 32, pp. 5–9, Aug. 1990.

[50] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, and R. Chou, "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," *Systematic Rev.*, vol. 10, no. 1, p. 89, Dec. 2021, doi: 10.1186/s13643-021-01626-4.

[51] M. Aria and C. Cuccurullo, "Bibliometrix: An R-tool for comprehensive science mapping analysis," *J. Informetrics*, vol. 11, no. 4, pp. 959–975, Nov. 2017, doi: 10.1016/j.joi.2017.08.007.

[52] J. Luengo, D. García-Gil, S. Ramírez-Gallego, S. García, and F. Herrera, *Big Data Preprocessing*. Cham, Switzerland: Springer, 2020, pp. 101–119.

[53] *Posit*. Accessed: Jun. 28, 2023. [Online]. Available: https://posit.co/downloads

[54] M. J. Cobo, A. G. López-Herrera, E. Herrera-Viedma, and F. Herrera, "Science mapping software tools: Review, analysis, and cooperative study among tools," (in English), *J. Amer. Soc. Inf. Sci. Technol.*, vol. 62, no. 7, pp. 1382–1402, Jul. 2011, doi: 10.1002/asi.21525.

[55] P. Pons and M. Latapy, "Computing communities in large networks using random walks," *J. Graph Algorithms Appl.*, vol. 10, no. 2, pp. 191–218, 2006, doi: 10.7155/jgaa.00124.

[56] A. M. Carrington, D. G. Manuel, P. W. Fieguth, T. Ramsay, V. Osmani, B. Wernly, C. Bennett, S. Hawken, O. Magwood, Y. Sheikh, M. McInnes, and A. Holzinger, "Deep ROC analysis and AUC as balanced average accuracy, for improved classifier selection, audit and explanation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 329–341, Jan. 2023, doi: 10.1109/TPAMI.2022.3145392.

[57] S. Wang and X. Yao, "Using class imbalance learning for software defect prediction," *IEEE Trans. Rel.*, vol. 62, no. 2, pp. 434–443, Jun. 2013, doi: 10.1109/Tr.2013.2259203.

[58] Microsoft. (2023). *Telemetry for Predictive Maintenance*. Accessed: Mar. 23, 2023. [Online]. Available: https://www.kaggle.com/datasets/arnabbiswas1/microsoft-azure-predictive-maintenance

[59] Pump. (2023). *Pump Sensor Data*. Accessed: Mar. 23, 2023. [Online]. Available: https://www.kaggle.com/datasets/nphantawee/pump-sensor-data

[60] M. Das, R. Cui, D. R. Campbell, G. Agrawal, and R. Ramnath, "Towards methods for systematic research on big data," in *Proc. IEEE Int. Conf. Big Data*, Oct. 2015, pp. 2072–2081. [Online]. Available: https://ieeexplore.ieee.org/document/7363989/

[61] I. Martinez, E. Viles, and I. G. Olaizola, "Data science methodologies: Current challenges and future approaches," *Big Data Res.*, vol. 24, May 2021, Art. no. 100183, doi: 10.1016/j.bdr.2020.100183.

[62] F. Kamalov and H. Sulieman, "Time series signal recovery methods: Comparative study," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Oct. 2021, pp. 1–5.

[63] L. Wijesekara and L. Liyanage, "Air quality data pre-processing: A novel algorithm to impute missing values in univariate time series," in *Proc. IEEE 33rd Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2021, pp. 996–1001.

[64] O. Caelen, "A Bayesian interpretation of the confusion matrix," *Ann. Math. Artif. Intell.*, vol. 81, nos. 3–4, pp. 429–450, Dec. 2017, doi: 10.1007/s10472-017-9564-8.

[65] S. Minaee, R. Kafieh, M. Sonka, S. Yazdani, and G. Jamalipour Soufi, "Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning," *Med. Image Anal.*, vol. 65, Oct. 2020, Art. no. 101794, doi: 10.1016/j.media.2020.101794.

[66] D. K. McClish, "Analyzing a portion of the ROC curve," *Med. Decis. Making*, vol. 9, no. 3, pp. 190–195, Aug. 1989, doi: 10.1177/0272989x8900900307.

[67] D. J. Hand and R. J. Till, "A simple generalisation of the area under the ROC curve for multiple class classification problems," *Machine Learn.*, vol. 45, no. 2, pp. 171–186, Nov. 2001, doi: 10.1023/A:1010920819831.

[68] L. S. Shapley, *A Value for N-Person Games*. Santa Monica, CA, USA: RAND Corporation, 1952.

[69] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 1–10.

[70] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, 1st ed. Victoria, BC, Canada: Leanpub, 2021.

MELVIN KISTEN received the B.Sc. degree (Hons.) in computer science from Sol Plaatje University (SPU), South Africa. He is currently pursuing the M.Sc. degree in computer science with North-West University (NWU), South Africa. He is also an M.Sc. Researcher with the Council for Scientific and Industrial Research (CSIR), South Africa. His primary research interests include software engineering and data science, with a particular emphasis on explainable artificial intelligence. His fervent engagement in various projects mirrors his commitment to these domains. This passion fuels his quest for deeper understanding, marking him as a dedicated scholar.

ABSALOM EL-SHAMIR EZUGWU received the B.Sc. degree in mathematics with computer science and the M.Sc. and Ph.D. degrees in computer science from Ahmadu Bello University, Zaria, Nigeria.

He is currently a Full Professor in computer science with the Unit for Data Science and Computing, North-West University, Potchefstroom, South Africa. He has contributed significantly to the academic community through the publication of numerous articles in internationally refereed journals, edited books, conference proceedings, and local journals. His research interests include artificial intelligence, swarm intelligence, and nature-inspired algorithm design, with a specific emphasis on computational intelligence and metaheuristic solutions for real-world global optimization problems. He is an active member of prominent organizations, such as Association for Computing Machinery (ACM), International Association of Engineers (IAENG), and Operations Research Society of South Africa (ORSSA). His dedication to advancing the field of computer science is evident in both his academic achievements and his ongoing contributions to cutting-edge research.

MICHEAL O. OLUSANYA (Member, IEEE) received the Ph.D. degree from the University of KwaZulu-Natal (UKZN), South Africa, in 2015. He is currently a Senior Lecturer with the Department of Computer Science and Information Technology, Sol Plaatje University (SPU), South Africa. His research interests include revolves around the application of metaheuristics and artificial intelligence techniques to solve real-life optimization problems, computational intelligence, and data analytics.

Appendix D – Supervision Agreement

The purpose of this agreement is to ensure mutual working relationship between the supervisor(s) and the student. The agreement also provides clarity of roles and responsibilities and ensures commitment and accountability of all parties involved.

Programme: Master of Science in Computer Science Research Entity/School: Computer Science and Information Systems

Faculty: Natural and Agricultural Sciences

The supervisor and the student will:

1. Establish agreed roles and clear processes to be maintained by both parties. In the case of joint supervision, all participants' roles need to be clarified.
2. Communicate regularly and as frequently as is reasonable to ensure steady progress towards the completion of the proposal and research product (dissertation or thesis).
3. Keep appointments, be punctual and respond timeously to messages.
4. Inform one another of any planned leave of absences as well as changes in personal circumstances that might have a negative impact on the research schedule.
5. Ensure that the research is conducted according to the procedures and the requirements of the relevant ethics committee.
6. Complete progress reports as requested by relevant faculty higher degrees committee.

The supervisor will:

1. Undertake to provide guidance for the student's research project in relation to the design and scope of the project, the relevant literature and information sources, research methods and techniques and methods of data analysis.
2. Have a responsibility to be accessible to the student.
3. Be prepared for the meeting with the student. This includes being up-to-date on the latest work in his/her area of expertise.
4. Assess written work and provide feedback within a timeframe jointly agreed at the beginning of the research.
5. Give advice that can help the student to improve his/her writing. This may include referrals to research support programmes including language training and academic writing. The supervisor will provide guidance on technical aspects and discipline-specific requirements.
6. Support the student in the production of a research report, dissertation or thesis. Provision should be allowed for adequate, mutually respectful, discussion around recommendations made.
7. Assist with the compilation of a written time schedule which outlines the expected completion dates of successive stages of the work.
8. Discuss the ownership of research conducted by the student in accordance with the university guidelines and rules on intellectual property, co-authorship and copyright.
9. Ensure that the research product is not plagiarised.
10. Ensure that the student is made aware in writing of the lack of progress and/or of any work that is below the set standards.

The student will:

1. Undertake to work independently under the guidance of the supervisor. This includes reading widely to ensure that the literature pertinent to his/her chosen topic has been identified and consulted.
2. Be obliged to make appointments to see the supervisor and will arrange meeting times well in advance.
3. Submit written work for discussion with the supervisor well in advance of a scheduled meeting. The kind and frequency of written work should be agreed with the supervisor at the outset of the research.
4. Take responsibility for the accuracy of language, the overall structure and coherence of the final research report, dissertation or thesis rests with the student.
5. Undertake to pay attention the advice given by the supervisor and to engage in discussion around suggestions made.
6. Take responsibility for the quality and presentation of the work.
7. Maintain a focus on his/her research area and to work within the agreed time schedule.
8. Honour agreements about ownership of the research and in accordance with the university's guidelines and rules in relation to co-authorship, copyright and intellectual property.
9. Ensure that the work contains no instances of plagiarism, and that all citations are properly referenced and the list of references is accurate, complete and consistent.
10. Work in accordance with the criteria of acceptability as supplied by the supervisor. This includes attending all support programmes (e.g. Seminars, short courses, etc.) That may be prescribed to improve performance.

We confirm that we have read and understood this statement and agree to be guided by its principles.

Name of student: Melvin Kisten

Student's signature: _____ Date: 01/02/2024

Name of supervisor: Prof. Absalom Ezugwu

Supervisor's signature: _____ Date: 01/02/2024

Name of Co-supervisor: Dr. Micheal Olusanya

Co-supervisor's signature: _____ Date: 01/02/2024

Appendix E – Progress report for Doctoral/Master's studies

Student name: **Melvin Kisten**

Supervisor: **Prof. Absalom Ezugwu**

Title: Explainable Machine Learning Model for Predictive Maintenance in Smart Agricultural Facilities

Date of first registration: 17/03/2023

Milestone	**Progress/achievement**	**Reasons for lack of Progress**	**Proposed mitigation/support**
Proposal	Completed		
Literature review	Completed		
Ethics approval	Completed		
Title registration	Completed		
Data collection	Completed		
Laboratory analysis	Completed		
Data analysis	Completed		
Thesis write-up	Completed		

Confirmation from student and supervisor **Date**

Student _____ 01/02/2024

Supervisor _____ 01/02/2024

Appendix F – Similarity Report

Explainable Machine Learning Model for Predictive Maintenance in Smart Agricultural Facilities

ORIGINALITY REPORT

7%	5%	4%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Liverpool John Moores University Student Paper	1%
2	www.medrxiv.org Internet Source	<1%
3	arxiv.org Internet Source	<1%
4	ipfs.io Internet Source	<1%
5	mdpi-res.com Internet Source	<1%
6	pure.tue.nl Internet Source	<1%
7	github.com Internet Source	<1%
8	www.mdpi.com Internet Source	<1%