

**FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO**

# **Hybrid Failure Prognosis Approach combining Data-Driven and Knowledge-Based Methods**

**Miguel Ramos Martins Rua Almeida**



Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Gil Manuel Magalhães de Andrade Gonçalves

July 29, 2024



# Resumo

Na indústria moderna, caracterizada por um aumento sem precedentes na geração de dados, a utilização eficaz dessa informação é crucial. No contexto industrial, as falhas de equipamento apresentam grandes desafios, resultando em perdas financeiras, desperdício de recursos e um ambiente insustentável. Esta dissertação aborda estes desafios, prevendo proativamente potenciais falhas de equipamento, de forma a mitigar os impactos adversos e promover um ambiente industrial mais sustentável e eficiente.

Uma abordagem híbrida é proposta, integrando metodologias baseadas em dados e conhecimento do domínio. Os métodos baseados em dados utilizam algoritmos de aprendizagem computacional para analisar extensos volumes de dados, de modo a identificar padrões, correlações e anomalias. Por sua vez, os métodos baseados em conhecimento, tiram proveito do conhecimento de especialistas da área. Ao combinar as duas abordagens, o sistema proposto visa melhorar a precisão e a fiabilidade dos sistemas de manutenção preditiva.

Aplicando técnicas de aprendizagem computacional e do conhecimento do domínio, esta dissertação apresenta um sistema de previsão e relatório de falhas. Seis algoritmos de aprendizagem computacional foram avaliados, com as previsões destes modelos ajustadas através de regras derivadas do conhecimento do domínio. Adicionalmente, foram incorporadas técnicas de Inteligência Artificial Explicável de forma a aumentar a interpretabilidade dos resultados dos modelos de previsão. Estas técnicas auxiliam os utilizadores do sistema a compreender as previsões geradas, capacitando-os com o conhecimento para personalizar o sistema, através da interface *web*, com as suas próprias regras. Os utilizadores podem também submeter relatórios de falhas, que desencadeiam o retreino do modelo, para facilitar uma melhoria contínua do sistema de previsão.

A integração de regras derivadas do conhecimento do domínio melhorou o desempenho dos modelos de previsão em até 28 pontos percentuais na métrica *F1 Score*, com a melhor abordagem híbrida a alcançar um *F1 Score* de 90% e um *Recall* de 92%. Esta dissertação contribui para o avanço da compreensão do estado dos equipamentos, permitindo a identificação proativa de potenciais falhas, promovendo a inovação, e incentivando práticas industriais mais eficientes e sustentáveis.

**Palavras-Chave:** Previsão de Falhas, Abordagens Híbridas, Métodos Baseados em Conhecimento, Métodos Baseados em Dados, Inteligência Artificial Explicável.



# Abstract

In modern industry, characterised by an unprecedented surge in data generation, effectively utilising this information is crucial. In the industrial landscape, equipment failures pose significant challenges, leading to financial losses, wasted resources, and an unsustainable environment. This dissertation addresses these challenges by proactively predicting potential equipment failures to mitigate adverse impacts and foster a more sustainable and efficient industrial environment.

A hybrid approach is proposed, integrating data-driven and knowledge-driven methodologies. Data-driven methods use Machine Learning (ML) algorithms to analyse extensive datasets, identifying patterns, correlations, and anomalies. Concurrently, knowledge-driven methods take advantage of the expertise of industry professionals to provide insights in addition to quantitative precision. By combining both approaches, the proposed system aims to improve the accuracy and reliability of predictive maintenance systems.

By leveraging ML techniques and domain knowledge, this dissertation presents a comprehensive failure prediction and reporting system. Six different ML algorithms were evaluated, with predictions refined using rule-based adjustments derived from domain knowledge. Additionally, Explainable Artificial Intelligence (XAI) techniques were incorporated to enhance the interpretability of the models. These techniques help the system's users understand the model's predictions, empowering them with the knowledge to customise the system via a web interface with their own rules. Users can also submit failure reports, which trigger model retraining and continuous improvement.

Integrating domain-specific rules improved performance by up to 28 percentage points in the F1 Score metric, with the best hybrid approach achieving an F1 Score of 90% and a recall of 92% in failure prediction. This adaptive hybrid approach enhances prediction accuracy and demonstrates the advantages of integrating hybrid approaches. This work contributes to advancing the understanding of equipment health, enabling proactive identification of potential failures and promoting innovation and more efficient, sustainable industrial practices.

**Keywords:** Failure Prediction, Hybrid Approaches, Knowledge-based Methods, Data-driven Methods, Explainable Artificial Intelligence.



# United Nation's Sustainable Development Goals

The United Nations' Sustainable Development Goals are a global call to action to end poverty, protect the planet, and ensure prosperity for all by 2030. These goals address various challenges, from economic inequality to environmental sustainability and social inclusivity. Each goal is interconnected, aiming to achieve a better and more sustainable future for all [1].

This work on a hybrid failure prognosis approach for metro trains aligns directly with Sustainable Development Goal (SDG) 9: Industry, Innovation, and Infrastructure, and significantly contributes to SDG 11: Sustainable Cities and Communities.

The project enhances infrastructure resilience through advanced predictive maintenance techniques. By preventing unexpected failures in metro trains, it improves the reliability and efficiency of critical urban infrastructure, a key objective of SDG 9.

Metro trains are vital for urban mobility, and ensuring their reliability supports the development of sustainable and resilient transport systems. This contributes to safer and more dependable public transportation, reduces environmental impact, and enhances the quality of life in urban areas, which are central goals of SDG 11.

## Specific Targets

### SDG 9: Industry, Innovation, and Infrastructure

- **Target 9.1:** Develop quality, reliable, sustainable, and resilient infrastructure to support economic development and human well-being.
- **Target 9.4:** Upgrade infrastructure to make it sustainable, with increased resource-use efficiency and the adoption of clean technologies.

### SDG 11: Sustainable Cities and Communities

- **Target 11.2:** Provide access to safe, affordable, accessible, and sustainable transport systems for all, improving road safety, notably by expanding public transport.
- **Target 11.6:** Reduce the adverse per capita environmental impact of cities, including by paying special attention to air quality and municipal and other waste management.

## Contribution

- **Improving Quality of Life:** Reliable and efficient public transportation reduces travel disruptions and ensures commuter safety.

- **Environmental Protection:** Predictive maintenance minimises the need for emergency repairs, which are often resource-intensive, thus promoting more sustainable urban operations.

## Performance Indicators

To measure the project's impact, the primary performance indicators focus on the system's predictive accuracy and reliability. These metrics evaluate how effectively the hybrid failure prognosis approach can foresee and prevent potential failures in metro trains.

A highly accurate and reliable system reduces train downtime by enabling timely maintenance before failures occur. This reduction enhances the overall efficiency of the metro system, ensuring trains remain operational and schedules are maintained.

Reduced downtime leads to fewer service interruptions, increasing passenger satisfaction. Improved reliability improves resource efficiency and supports environmental sustainability by minimising unnecessary repairs and replacements.

In summary, the project's impact is primarily measured through system performance indicators, with broader benefits including increased passenger satisfaction, improved resource efficiency, and a positive environmental impact.

# Agradecimentos

Gostaria de agradecer ao meu orientador, Professor Gil Gonçalves, pela confiança depositada em mim e por toda a ajuda ao longo deste projeto.

Também gostaria de agradecer ao meu coorientador, Eliseu Pereira, pelo apoio e orientação ao longo deste projeto, e por estar sempre pronto a ajudar.

Obrigado ao Tomás e ao Diogo por me apoiarem diariamente.

A todos os amigos que fiz na faculdade, em particular ao Duarte, à Inês, ao Miguel, ao Afonso, ao Pedro, à Maria e à Cátia, obrigado por tornarem este percurso muito mais fácil e pelo apoio constante ao longo destes cinco anos. Sei que, sem vocês, esta experiência não teria sido a mesma. Em especial, obrigado ao Afonso por ter percorrido este caminho comigo, desde o infantário até ao último dia na faculdade. Não poderia ter pedido melhor companheiro.

À minha namorada, Beatriz, obrigado por seres uma fonte constante de apoio, alegria e inspiração.

Gostaria de expressar um agradecimento especial aos meus pais, Anabela e Carlos, por estarem sempre prontos a ajudar-me e por serem os melhores modelos que poderia ter. O vosso apoio incondicional e os valores que me transmitiram foram essenciais para a pessoa que sou hoje. Agradeço também à minha irmã Rita pelo seu apoio constante.

Miguel Almeida



*“Believe you can and you’re halfway there.”*

Theodore Roosevelt



# Contents

|          |                                                            |           |
|----------|------------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                        | <b>1</b>  |
| 1.1      | Context . . . . .                                          | 1         |
| 1.2      | Challenges and Limitations of Current Approaches . . . . . | 2         |
| 1.3      | Objectives . . . . .                                       | 3         |
| 1.4      | Dissertation Structure . . . . .                           | 3         |
| <b>2</b> | <b>Literature Review</b>                                   | <b>5</b>  |
| 2.1      | Research Methods . . . . .                                 | 5         |
| 2.2      | Background . . . . .                                       | 6         |
| 2.2.1    | Industry 4.0 . . . . .                                     | 6         |
| 2.2.2    | Predictive Maintenance . . . . .                           | 7         |
| 2.2.3    | Process Flow . . . . .                                     | 8         |
| 2.3      | Pre-Processing . . . . .                                   | 9         |
| 2.4      | Data-driven Methods . . . . .                              | 10        |
| 2.4.1    | Supervised Methods . . . . .                               | 11        |
| 2.4.2    | Unsupervised Methods . . . . .                             | 12        |
| 2.4.3    | Methods Comparison . . . . .                               | 12        |
| 2.4.4    | Conclusion . . . . .                                       | 13        |
| 2.5      | Knowledge-driven Methods . . . . .                         | 14        |
| 2.6      | Hybrid Approaches . . . . .                                | 14        |
| 2.7      | Performance Metrics . . . . .                              | 17        |
| 2.8      | Model Explainability . . . . .                             | 18        |
| 2.8.1    | Explainable Artificial Intelligence . . . . .              | 19        |
| 2.9      | Summary and Gap Analysis . . . . .                         | 22        |
| <b>3</b> | <b>Implementation</b>                                      | <b>25</b> |
| 3.1      | System Architecture . . . . .                              | 25        |
| 3.2      | Dataset . . . . .                                          | 27        |
| 3.3      | Data Pre-processing . . . . .                              | 29        |
| 3.4      | Failure Prediction Model . . . . .                         | 31        |
| 3.4.1    | Data-driven Approach . . . . .                             | 31        |
| 3.4.2    | Knowledge-driven Approach . . . . .                        | 32        |
| 3.5      | Explainability Incorporation . . . . .                     | 36        |
| 3.6      | Application . . . . .                                      | 37        |
| 3.6.1    | "Dashboard" Page . . . . .                                 | 38        |
| 3.6.2    | "Failure Report" Page . . . . .                            | 39        |
| 3.6.3    | "Add Failure Report" Page . . . . .                        | 39        |
| 3.6.4    | "Rules" Page . . . . .                                     | 40        |

|          |                                           |           |
|----------|-------------------------------------------|-----------|
| 3.6.5    | "Add New Rule" Page . . . . .             | 41        |
| 3.6.6    | "Model Explanation" Page . . . . .        | 42        |
| 3.7      | Summary . . . . .                         | 43        |
| <b>4</b> | <b>Validation and Discussion</b>          | <b>45</b> |
| 4.1      | Data-driven Models Performance . . . . .  | 45        |
| 4.2      | Rule-Based Adjustments . . . . .          | 47        |
| 4.3      | Model Explainability . . . . .            | 50        |
| 4.4      | Graphical User Interface . . . . .        | 51        |
| 4.5      | Prediction and Explanation Time . . . . . | 54        |
| 4.6      | System Validation . . . . .               | 55        |
| 4.7      | Summary . . . . .                         | 56        |
| <b>5</b> | <b>Conclusion and Future Work</b>         | <b>59</b> |
|          | <b>References</b>                         | <b>61</b> |
| <b>A</b> | <b>Conference Paper</b>                   | <b>65</b> |

# List of Figures

|      |                                                                                                                                         |    |
|------|-----------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1  | Primary categories in the realm of maintenance. . . . .                                                                                 | 7  |
| 2.2  | Process flow of the failure prediction process. . . . .                                                                                 | 9  |
| 2.3  | Data-driven Methods. . . . .                                                                                                            | 11 |
| 2.4  | Combination Strategies for Hybrid Approaches [2]. . . . .                                                                               | 15 |
| 2.5  | Model Explainability techniques [3]. . . . .                                                                                            | 19 |
| 3.1  | Proposed system architecture. . . . .                                                                                                   | 26 |
| 3.2  | Image of a "Metro do Porto" metro train (retrieved from [4]). . . . .                                                                   | 27 |
| 3.3  | Correlation matrix calculated using Pearson correlation coefficient. . . . .                                                            | 30 |
| 3.4  | Scree plot showing the explained variance ratio by number of components in PCA. . . . .                                                 | 31 |
| 3.5  | Time series plots generated during the Exploratory Data Analysis. . . . .                                                               | 34 |
| 3.6  | Kernel Density Estimation plots illustrating data distribution characteristics, generated during the Exploratory Data Analysis. . . . . | 35 |
| 3.7  | UI mockup of the "Dashboard" page. . . . .                                                                                              | 38 |
| 3.8  | UI mockup of "Failure Report" page. . . . .                                                                                             | 39 |
| 3.9  | UI mockup of "Add Failure Report" page. . . . .                                                                                         | 40 |
| 3.10 | UI mockup of "Rules" page. . . . .                                                                                                      | 41 |
| 3.11 | UI mockup of "Add New Rule" page. . . . .                                                                                               | 42 |
| 3.12 | UI mockup of "Model Explanation" page. . . . .                                                                                          | 43 |
| 4.1  | Confusion matrix of XGBoost model. . . . .                                                                                              | 46 |
| 4.2  | Confusion matrix of Random Forest model, before the implementation of rules. . . . .                                                    | 48 |
| 4.3  | Confusion matrix of Random Forest model, after the implementation of rules. . . . .                                                     | 49 |
| 4.4  | An example of a LIME explanation for a failure prediction. . . . .                                                                      | 50 |
| 4.5  | "Dashboard" page, in the web application. . . . .                                                                                       | 51 |
| 4.6  | "Failures Reports" page, in the web application. . . . .                                                                                | 52 |
| 4.7  | "Add Failure Reports" page, in the web application. . . . .                                                                             | 52 |
| 4.8  | "Rules" page, in the web application. . . . .                                                                                           | 53 |
| 4.9  | "Add New Rules" page, in the web application. . . . .                                                                                   | 53 |
| 4.10 | "Model Explanation" page, in the web application. . . . .                                                                               | 54 |
| 4.11 | "Model Explanation" page, with open pop-up. . . . .                                                                                     | 54 |



# List of Tables

|     |                                                                                                                                 |    |
|-----|---------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 | Pre-processing techniques employed in the literature. . . . .                                                                   | 10 |
| 2.2 | Best data-driven algorithm for each application. . . . .                                                                        | 13 |
| 2.3 | Summary of the reviewed papers. . . . .                                                                                         | 17 |
| 2.4 | Performance metrics utilised in the papers reviewed in [5]. . . . .                                                             | 18 |
| 2.5 | Overview of the related research. . . . .                                                                                       | 23 |
| 3.1 | Failure reports provided by field-experts [6]. . . . .                                                                          | 28 |
| 4.1 | Results of the considered evaluation metrics of the implemented Machine Learning models. . . . .                                | 45 |
| 4.2 | Comparison of the implemented Machine Learning models performance before and after incorporating knowledge-based rules. . . . . | 47 |
| 4.3 | Evaluation metrics of rule application in Machine Learning models. . . . .                                                      | 49 |
| 4.4 | Time taken for various system tasks. . . . .                                                                                    | 55 |



# Acronyms and Symbols

|          |                                                   |
|----------|---------------------------------------------------|
| AI       | Artificial Intelligence                           |
| ANN      | Artificial Neural Networks                        |
| AUC      | Area Under the Curve                              |
| BT       | Boosting Tree                                     |
| CHAID    | General Chi-square Automatic Interaction Detector |
| DTs      | Decision Trees                                    |
| EDA      | Exploratory Data Analysis                         |
| ELI5     | Explain Like I'm 5                                |
| FDE      | Failure Diagnosis Explainability                  |
| GB       | Gradient Boosting                                 |
| GBM      | Gradient Boosting Machines                        |
| Grad-CAM | Gradient-weighted Class Activation Mapping        |
| KDE      | Kernel Density Estimation                         |
| LIME     | Local Interpretable Model-agnostic Explanations   |
| LR       | Logistic Regression                               |
| LSTM     | Long Short-Term Memory                            |
| ML       | Machine Learning                                  |
| MQTT     | Message Queuing Telemetry Transport               |
| PCA      | Principal Component Analysis                      |
| RF       | Random Forest                                     |
| ROC      | Receiver Operating Characteristic Curve           |
| RQs      | Research Questions                                |
| RUL      | Remaining Useful Life                             |
| SDG      | Sustainable Development Goal                      |
| SHAP     | SHapley Additive exPlanations                     |
| SVM      | Support Vector Machine                            |
| XAI      | Explainable Artificial Intelligence               |
| XGB      | XGBoost                                           |



# Chapter 1

## Introduction

Chapter 1 presents an overview of the dissertation, covering its context in Section 1.1, addressing challenges in Section 1.2 and defining objectives in Section 1.3. Finally, it outlines the document structure in Section 1.4.

### 1.1 Context

In the context of industrial operations, equipment failures pose a critical challenge, not only as a substantial drain on time but also as a significant financial burden. As industries have evolved, so have maintenance strategies, with a notable shift from reactive to proactive approaches. An essential component of this evolutionary process has been the development of predictive maintenance, which is the technique of anticipating possible equipment faults before they happen.

Historically, industries often suffered the consequences of unexpected equipment breakdowns, leading to unscheduled downtime, increased maintenance expenses, and lost productivity. The transition to predictive maintenance marked a significant stride, using a proactive approach that uses technology to anticipate potential failures, thereby mitigating the disruptive effects associated with unplanned downtimes.

Industry 4.0, often called the fourth industrial revolution, is revolutionising the manufacturing and industrial processes. This era is characterised by the integration of modern technology, data-driven insights, and smart automation to improve productivity, efficiency, and overall operational intelligence. Key components of Industry 4.0 include the Internet of Things, which makes it easier to link devices and sensors, providing real-time data for decision-making. Additionally, Artificial Intelligence (AI) and machine learning are essential, enabling systems to learn from data patterns, predict outcomes, and optimise processes autonomously.

While generating a substantial volume of data may seem advantageous, it introduces certain challenges. The rise of data in industrial settings presents both challenges and opportunities. On one hand, there is a wealth of information that, if harnessed effectively, can revolutionize how

industries approach equipment management. On the other hand, the sheer volume and complexity of this data necessitate innovative methodologies for meaningful interpretation and actionable insights.

Within the landscape of analytical solutions, differentiating methodologies emerge, notably in the form of data-driven methods and knowledge-driven methods. The first one relies on the assimilation and analysis of datasets retrieved from sensors. In contrast, the latter utilises the insights and expertise of domain-specific professionals.

Data-driven methods, through advanced algorithms and statistical models, analyse extensive datasets to uncover patterns, correlations, and anomalies. This approach is effective in understanding the specifics of manufacturing, aiding in checking the health of equipment, and predicting potential failures.

Knowledge-driven methods rely on the expertise of industry professionals and draw from experiential and contextual insights in the manufacturing domain. This integration of expertise and knowledge into the analytical process ensures a more informed approach to addressing challenges within the manufacturing landscape.

Nevertheless, the utilisation of these methods in isolation has often resulted in a potential loss of information. By integrating these two approaches, it unlocks a range of advantages. This combination not only enhances the overall robustness of analytical solutions but also enables a more comprehensive understanding of the subject at hand. Moreover, fosters a culture of continuous improvement, promoting sustainable practices in industrial operations. The collaboration between each one capitalises on the strengths of each, forming a cohesive and efficient analytical framework.

## **1.2 Challenges and Limitations of Current Approaches**

While the integration of advanced technologies and data-driven insights holds tremendous promise for revolutionising industrial operations, it is not without its challenges. The advantages of having access to large amounts of data are undeniable. However, there are significant obstacles to the efficient use of this data due to its immense quantity and complexity. Sophisticated approaches and analytical tools are needed to extract relevant information from the massive amount of data. The risk lies in the potential for information overload, where the abundance of data may lead to confusion rather than clarity.

Furthermore, the complexity of industrial processes and the large amount of data from different sensors make the complexity substantially higher. Traditional machine-learning approaches may struggle to understand the data at hand, limiting the efficiency of the decision-making processes. As a result, there is a growing demand for innovative approaches that are capable of managing this complexity and providing useful insights.

Another significant challenge lies in the temporal aspect of data. Industrial processes are dynamic, with conditions and variables changing over time. Static analytical models may have

difficulty capturing the dynamic nature of these processes, leading to predictions and insights that may not align with real-time scenarios.

Moreover, the utilisation of singular methodologies, whether data-driven or knowledge-driven, presents limitations in the understanding of equipment health. Also, relying only on data-driven methods may overlook critical insights from industry experts, while an exclusive use of knowledge-driven approaches may miss patterns and anomalies detectable through advanced algorithms.

In navigating these challenges, the quest for effective solutions becomes imperative. The integration of data-driven and knowledge-driven methodologies emerges as a strategic response to the limitations posed by individual approaches. By combining the best features of both approaches, a hybrid strategy aims to provide a more comprehensive understanding of industrial processes. Recognising and resolving these issues becomes critical as industries work to utilise well the abundance of data at their disposal.

### 1.3 Objectives

As previously stated, relying on a singular method can prove disadvantageous in the field of failure prediction. Therefore, the main objective of this dissertation is to implement a hybrid approach integrating both data-driven and knowledge-driven methodologies for the prediction of potential failures. This approach seeks to enhance not only predictive accuracy but also aims to offer a more comprehensive and improved understanding of the numerous factors influencing equipment health and failure prognostication.

In the prediction process, data-driven methods will be deployed to effectively handle information from diverse sensors, exploiting the inherent advantages of these methods. Moreover, to counteract information loss, expert knowledge will be incorporated through knowledge-driven methods. This integration aims to establish a cohesive analytical framework that maximises the strengths of both approaches, ensuring a more comprehensive and nuanced understanding.

This dissertation integrates with the principles of Industry 4.0 by capitalising on modern technology, including leveraging data-driven insights and smart automation, contributing to the evolution of industrial practices. Also, in an era where sustainability is critical, this research promotes environmentally conscious practices by minimising resource wastage, mitigating the negative impact of equipment failures on sustainability, and promoting sustainable practices within the industry.

### 1.4 Dissertation Structure

This dissertation is organised as follows: Chapter 2 presents the fundamental knowledge and literature review. Chapter 3 details the methodology employed to develop and implement the failure prediction and reporting system, while Chapter 4 presents the validation and discussion of the results. Lastly, Chapter 5 summarises the main contributions and conclusions of this dissertation, alongside directions for future work.



## Chapter 2

# Literature Review

Chapter 2 reviews the current state-of-the-art. Firstly, Section 2.1 begins by detailing the research methods employed. Following this, Section 2.2 provides essential background knowledge relevant to the engineering context of this work. The following sections cover various topics: Section 2.3 addresses pre-processing, Section 2.4 explores data-driven methods, Section 2.5 examines knowledge-driven methods, Section 2.6 discusses hybrid approaches, and Section 2.7 reviews performance metrics. Additionally, Section 2.8 focuses on model explainability, highlighting Explainable Artificial Intelligence. Finally, Section 2.9 summarises the chapter and includes a gap analysis.

### 2.1 Research Methods

A systematic review of the hybrid failure prognosis approach combining data-driven and knowledge-based methods was conducted. This led to the formulation of the following research questions (RQs):

**RQ1:** How can expert knowledge be effectively integrated into data-driven predictive maintenance models to enhance accuracy and reliability?

**RQ2:** What role does the integration of expert knowledge play in achieving a more sustainable and environmentally friendly predictive maintenance system?

**RQ3:** How does the combination of data-driven and knowledge-based methods impact the overall efficiency and effectiveness of equipment maintenance and replacement strategies?

**RQ4:** What challenges and opportunities arise in the implementation of a hybrid predictive maintenance system that considers both expert knowledge and data-driven methodologies?

The PICO method was employed to identify keywords and formulate search strings based on the research questions. The identified keywords, including data-driven methods, knowledge-based methods, hybrid, failure prediction, Industry 4.0, XAI, and predictive maintenance, were then organised into sets and used to construct the search string.

- **Set 1:** Scoping the search for industry 4.0, i.e. “industry 4.0”.
- **Set 2:** Search terms directly related to failure prediction, e.g. “failure prediction”, and “failure prognosis”.
- **Set 3:** Scoping the search for predictive maintenance, i.e. “predictive maintenance”.
- **Set 4:** Search terms directly related to hybrid approaches, e.g. “hybrid approaches”, and “hybrid”.
- **Set 5:** Scoping the search for data-driven methods, i.e. “data-driven methods”.
- **Set 6:** Search terms directly related to Knowledge-based methods, e.g. “Knowledge-based methods”, and “expert knowledge”.
- **Set 7:** Search terms directly related to Explainable Artificial Intelligence, e.g. “Explainable Artificial Intelligence”, and “XAI”.

The identified keywords were used to perform searches in online databases such as Google Scholar and Scopus. In addition, relevant studies were included using a backward snowballing sampling approach. The obtained results were then filtered using an exclusion criteria:

- Studies not presented in English.
- Studies presenting non-peer-reviewed material.
- Studies not accessible in full text.

## **2.2 Background**

### **2.2.1 Industry 4.0**

The manufacturing industries are undergoing rapid change, and Industry 4.0 has brought out a fundamental shift driven by the escalating volume of data generated within these environments. Industry 4.0 has a significant impact because it strongly focuses on automation, connection, and the intelligent utilisation of data to bring industrial processes into a new era of efficiency and predictability.

At the heart of this transformative era is the integration of advanced data analytics solutions, and notably, machine learning algorithms play a pivotal role, especially when assessing health and predicting potential equipment failures in the industrial sector. As Zonta Tiago et al. said, "Data

is the key to this generation of information that can anticipate or collaborate in making predictive decisions." [7].

These algorithms are not only capable of interpreting the details of machine health, but they also serve as prognosticators, surpassing conventional techniques in their ability to properly identify future problems. The combination of industrial machinery and data analytics has resulted in a new era of predictive ability where decisions are made based on data insights that minimise downtime and extend the life of critical equipment.

### 2.2.2 Predictive Maintenance

In the realm of maintenance, four primary categories have been identified, as outlined by Zonta et al. [7] and shown in Figure 2.1: corrective, preventive, predictive, and prescriptive strategies. Corrective maintenance comes into play post-incident, addressing issues after they have occurred. On the other hand, preventive maintenance takes a proactive approach through scheduling tasks, routine inspections, part replacements, and predefined actions to prevent potential problems. Predictive maintenance anticipates when maintenance is required based on the current condition of the equipment. Using real-time data and advanced analytics, it predicts potential failures, enabling timely and precise interventions. Lastly, prescriptive maintenance recommends specific actions to optimise equipment performance. This category offers recommendations for the most effective maintenance strategies, considering cost, efficiency, and resource optimisation.

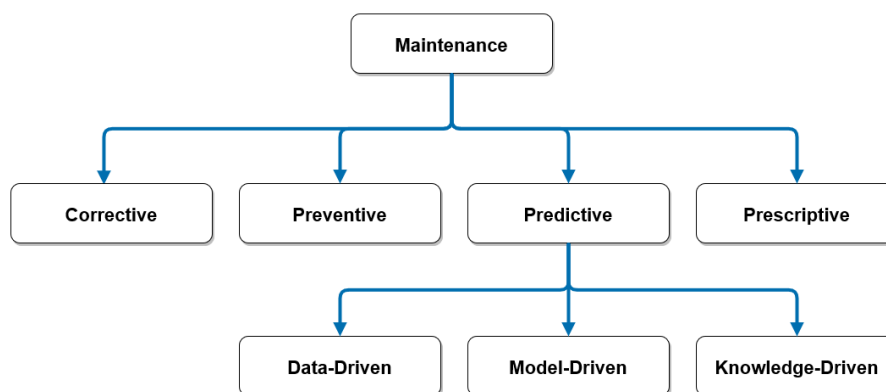


Figure 2.1: Primary categories in the realm of maintenance.

Predictive maintenance represents a pivotal evolution in the management of industrial assets, offering a proactive and strategic approach to equipment reliability. In essence, it involves the use of advanced data analytics, ML, and sensor technologies to predict potential failures in machinery before they occur. This departure from traditional reactive maintenance practices empowers industries to not only address issues before they escalate but also optimise operational efficiency and extend the lifespan of critical equipment.

In the landscape of predictive maintenance, three prominent classification approaches for prediction emerge, each distinguished by its methodology and underlying principles: data-driven, model-driven, and knowledge-driven methods.

Data-driven methods form the backbone of predictive maintenance, utilising extensive datasets sourced from sensors, PLCs, and other industrial data streams. These methodologies employ advanced algorithms and statistical models to analyse historical and real-time data, unveiling complex patterns, correlations, and anomalies.

Model-based methods centre around creating explicit mathematical or computational models of industrial systems. These models encapsulate the relationships governing machinery, offering insights into potential failure modes. By simulating equipment behaviour under different conditions, these methods provide proactive tools for forecasting and optimising maintenance strategies.

Knowledge-based methods in predictive maintenance draw on the expertise of industry professionals, integrating experiential knowledge into the prediction process. Unlike data-driven or model-based methods, these approaches capture human insights through rule-based systems or expert systems. Industry professionals contribute qualitative understanding, particularly in scenarios requiring contextual insights and rare event considerations. By enriching predictions with human expertise, knowledge-based methods provide a more informed and nuanced approach to predictive maintenance.

### 2.2.3 Process Flow

The failure prediction process, as elucidated in [5] and illustrated in Figure 2.2, encompasses four key steps. In the initial phase, data is collected from diverse sources, including sensors, historical records, and operational logs. This data collection forms the foundational basis for subsequent analyses.

Subsequently, the extracted data undergoes a crucial phase of feature identification, where relevant attributes reflecting the machine's health and performance are carefully selected. This pre-processing step is pivotal for enhancing the model's capability to discern patterns and relationships within the data.

Following feature identification, an ML model is meticulously constructed, utilising advanced algorithms. The model is trained on historical data to learn the complex patterns associated with machine behaviour and potential failure modes.

Finally, the failure prediction process involves evaluating the performance of the constructed model. This evaluation is essential to assess the model's predictive accuracy and its ability to generalise to new, unseen data. Iterative refinement and validation processes may be employed to fine-tune the model, ensuring its effectiveness in real-world scenarios.

In essence, this four-step process represents an approach to utilising data-driven methodologies for failure prediction, providing a proactive framework for addressing potential issues and optimising maintenance strategies in industrial settings.

The subsequent sections of this chapter explore the aspects of process flow. Firstly, the particulars of pre-processing will be explored. Following that, the focus will shift towards an in-depth

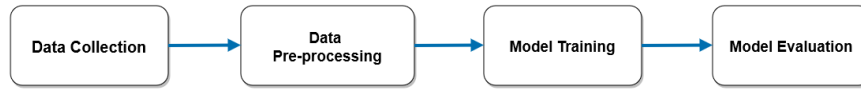


Figure 2.2: Process flow of the failure prediction process.

discussion on both data-driven and knowledge-driven methodologies. Subsequently, hybrid approaches and classification methods will be examined. Lastly, the chapter will conclude with an exploration and discussion of model explainability.

## 2.3 Pre-Processing

Data pre-processing plays a pivotal role in failure prediction within the field of ML, ensuring that raw data is transformed and optimised to enhance the performance of predictive models. The pre-processing stage can involve a series of steps aimed at cleaning, organising, and preparing the data for effective utilisation in the training and evaluation of predictive models. Additionally, the application of domain knowledge can improve the pre-processing process. The typical steps involved in this phase include:

- **Data Cleaning:** This step involves identifying and handling missing or wrong data points, ensuring the dataset is free from inconsistencies and inaccuracies.
- **Feature Engineering:** Creating new features or transforming existing ones can enhance the model's ability to capture patterns and relationships in the data.
- **Outlier Detection and Removal:** Identifying and handling outliers helps prevent them from wrongly influencing the model.
- **Normalisation:** Transforming variables to conform to a standard scale ensures consistent and fair treatment across different features.
- **Temporal Aggregation:** Aggregating data over time intervals can reveal patterns and trends, which is especially valuable in failure prediction scenarios.
- **Noise Reduction:** Filtering out irrelevant or noisy information from the dataset improves the model's ability to focus on meaningful patterns.
- **Dimensionality Reduction:** Techniques such as Principal Component Analysis (PCA) or feature selection help manage large datasets and reduce the risk of overfitting.

In Table 2.1, various studies are presented along with the pre-processing techniques they have employed.

Table 2.1: Pre-processing techniques employed in the literature.

| Study             | Feature Selection             |
|-------------------|-------------------------------|
| Li et al. [8]     | PCA                           |
| Canizo et al. [9] | PCA, correlation              |
| Leahy et al. [10] | PCA, variance, missing values |
| Susto et al. [11] | Discard constant variables    |
| Hu et al. [12]    | Domain knowledge features     |

From the information presented, it is apparent that a variety of pre-processing techniques have been employed, with Principal Component Analysis being the most commonly utilised. PCA is frequently employed to reduce the dimensionality of data and extract essential features for enhanced model performance. Notably, expert knowledge is often essential in the identification of pertinent features within this technique, underscoring the value of domain expertise in the pre-processing phase.

In [12], the used dataset contains operational data and associated features obtained from a wind turbine. This dataset is augmented with supplementary features derived from the original operational data using domain-specific knowledge.

## 2.4 Data-driven Methods

Data-driven methods in predictive maintenance represent a modern approach utilising the abundance of data generated in industrial settings. The primary goal is to identify patterns, correlations, and anomalies within the data, enabling the prediction of equipment health and potential failures.

A notable advantage of data-driven methods is their capacity to uncover complex relationships that may be challenging or impossible for human perception. By analysing diverse datasets, data-driven methods contribute to a deeper understanding of manufacturing nuances, facilitate equipment health assessments, and enhance the prediction of potential failures.

The application of data-driven methods goes beyond mere prediction, contributing to a comprehensive understanding of equipment conditions. By processing large volumes of data, these methods uncover details that might escape conventional analyses. Consequently, it facilitates a proactive and targeted approach to maintenance, minimising downtime, optimising resource utilisation, and contributing to overall operational efficiency.

In the context of data-driven methods for predictive maintenance, a fundamental distinction exists between supervised and unsupervised learning approaches as elucidated in [13] and illustrated in Figure 2.3, each contributing uniquely to the evolving landscape of industrial analytics.

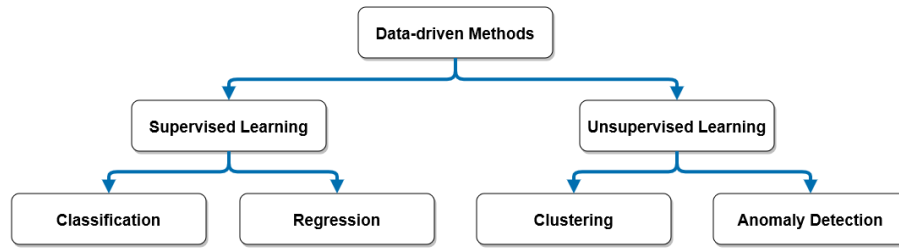


Figure 2.3: Data-driven Methods.

### 2.4.1 Supervised Methods

Supervised methods involve training predictive models on labelled datasets, where historical data is paired with known outcomes. These models learn patterns and relationships between input features and the corresponding target variables, enabling them to make predictions on new, unseen data. In predictive maintenance, supervised learning techniques such as classification and regression are widely employed [14]. Classification models can predict discrete outcomes, such as whether equipment will fail or not, while regression models provide quantitative estimates, such as the Remaining Useful Life (RUL) of machinery. In the following subsection, it will be presented the most commonly used ML algorithms for failure prediction in industrial maintenance, as outlined by [5].

- **Random Forest (RF):** Powerful learning algorithm widely used in ML. It operates by constructing multiple Decision Trees (DTs) during training and outputs the mode of the classes (classification) or the mean prediction (regression) of the individual trees.
- **Support Vector Machine (SVM):** Versatile supervised learning algorithm used for classification and regression tasks [15]. It works by finding a hyperplane that best separates data into distinct classes while maximising the margin between them. SVM is effective in high-dimensional spaces, making it valuable for various applications.
- **Artificial Neural Networks (ANN):** Fundamental component of ML, inspired by the structure and functioning of the human brain [15]. Comprising interconnected nodes organised in layers, ANNs excel in tasks like pattern recognition, classification, and regression. During training, the network adjusts its parameters to learn complex patterns and relationships within data, enabling it to make accurate predictions on new, unseen inputs.
- **Gradient Boosting (GB):** Powerful machine-learning technique designed for predictive modelling and regression analysis. It operates by combining the strength of multiple DTs to create a robust and accurate predictive model. The algorithm sequentially corrects errors made by its predecessors, placing more emphasis on instances where predictions fall short.

This iterative process allows Gradient Boosting to refine its model and improve overall predictive performance. Widely used for various tasks, including classification and regression, Gradient-boosting-based algorithms, such as XGBoost (XGB), have become instrumental in achieving state-of-the-art results in diverse ML applications.

- **Logistic Regression (LR):** Simple and widely used statistical method in ML for binary classification. It predicts the probability of an event occurring, transforming the output into values between 0 and 1. This simplicity makes LR a popular choice for various applications, with the goal of categorising data into two classes.

### 2.4.2 Unsupervised Methods

Unsupervised methods, on the other hand, operate without labelled data, making them particularly valuable in scenarios where the outcomes are not well-defined or unknown. Clustering and Anomaly Detection are common unsupervised tasks applied in predictive maintenance. Clustering methods group similar instances together, aiding in the identification of distinct operational patterns within datasets. Anomaly Detection, crucial for flagging irregularities in equipment behaviour, stands out in scenarios where unexpected events can have significant consequences. One example of an unsupervised method is the K-Means algorithm.

### 2.4.3 Methods Comparison

In [16], five algorithms, including Neural Network, SVM, RF algorithm, Boosting Tree (BT) algorithm, and General Chi-Square Automatic Interaction Detector (CHAID) algorithm, were employed to predict wind turbine faults. Among all the algorithms examined, the RF algorithm demonstrated the highest accuracy.

Furthermore, the research conducted in [17], focused on proactively detecting trucks in need of imminent maintenance, enabling scheduled component replacement before failure. A comparative analysis between RF and Long Short-Term Memory (LSTM) was performed, revealing that, despite the LSTM predictions displaying greater stability over time, RF achieved marginally superior Area Under the Curve (AUC) results.

In the study conducted in [18], a ML methodology employing SVM was applied for predicting faults in a centrifugal pump within the oil and gas industry. The SVM model successfully classified potential faults with a prediction accuracy of 98.1%. Similarly, in the research by Li et al. [8], an SVM was employed to enhance rail network speed by mitigating track failures. The performance of the SVM model was compared to that of an RF algorithm, with SVM demonstrating superior results in reducing derailments.

In the study detailed in [19], a specific type of Recurrent Neural Network, namely LSTM, was employed. LSTMs are particularly advantageous for learning from historical sequences and predicting subsequent events, as highlighted in the [19]. The implemented network was configured with a range of 50 to 150 cells, utilising the Adam optimiser and binary cross-entropy as the loss

function over 30 epochs. The training data comprised previous events extracted from machine failure logs. Notably, the algorithm exhibited robust performance, achieving a training loss exceeding 99%.

Moreover, in the study documented in [20], real data from vending machines was utilised to validate a proposed framework employing three distinct classifiers: SVM, RF, and Gradient Boosting Machines (GBM). Simulation results demonstrated that the diagnostics model achieved an accuracy exceeding 80%, establishing its capabilities as a robust choice for prognostic monitoring.

In a study by [21], the emphasis was on predicting downtime in a printing machine through real-time forecasts of potential failures. Historical machine data was employed to train ML classification algorithms, including RF, XGB, and LR. The findings revealed that, while all algorithms performed similarly in terms of the Receiver Operating Characteristic curve (ROC), RF and XGB surpassed LR in the F1 Score.

Table 2.2 summarises the most effective algorithms tested for the specific problem.

Table 2.2: Best data-driven algorithm for each application.

| Problem                             | Best ML Algorithm | References          |
|-------------------------------------|-------------------|---------------------|
| Wind turbine faults                 | RF                | Kusiak et al. [16]  |
| Air compressor failures             | RF                | Chen et al. [17]    |
| Centrifugal pump failures           | SVM               | Orrù et al. [18]    |
| Track failures                      | SVM               | Li et al. [8]       |
| Machine failures                    | LSTM              | Alves et al. [19]   |
| Vending machine components failures | GBM               | Xiang et al. [20]   |
| Printing machine failures           | RF, XGB           | Binding et al. [21] |

#### 2.4.4 Conclusion

In summary, the mentioned algorithms can be categorised into two broader groups, emphasising the crucial distinction between supervised and unsupervised learning approaches in the data-driven predictive maintenance landscape. Supervised methods, including RF, SVM, ANN, GB, and LR, utilise labelled datasets for training predictive models, making them highly effective at making predictions based on historical data. These methods are widely used for classification, regression, and remaining useful life estimation tasks.

The best-performing algorithms across the review studies were RF, SVM and Boosting-based algorithms, demonstrating effectiveness in fault prediction. It's important to note that the speed of machine-learning algorithms must be sufficiently fast to make timely fault predictions. Occasionally, an algorithm with superior accuracy may be less favourable if it consumes a considerable

amount of time for prediction, potentially impacting the ability to intervene and prevent machine failures promptly. Therefore, achieving an optimal balance between time efficiency and accuracy is crucial in practical applications.

## 2.5 Knowledge-driven Methods

Knowledge-driven methods in AI and ML play a pivotal role in decision-making by exploiting expert knowledge and domain expertise. These methods stand in contrast to data-driven approaches, as they involve the incorporation of explicit rules, ontologies, and logical reasoning into the decision-making process.

One popular example of knowledge-based methods is the development of Rule-Based Systems, where predefined rules guide decision-making [22]. These rules can be derived from expert knowledge, logical reasoning, or established guidelines within a specific domain. The transparency of rule-based systems makes them valuable in applications where interpretability is crucial.

Ontologies, structured representations of knowledge defining relationships and entities within a domain, are frequently used in knowledge-based methods [23]. They facilitate effective organisation and formalisation of information, contributing to more robust reasoning and decision-making.

Expert Systems are a specialised category of knowledge-based methods designed to emulate the decision-making capabilities of human experts. These systems integrate explicit rules and knowledge to provide solutions or recommendations based on input data, which is instrumental in areas where domain expertise is critical. However, traditional knowledge representations built upon expert systems demand a specific data structure design, and the majority of these systems possess complex architectures, restricting the ease of knowledge sharing and reuse [23].

One notable advantage of knowledge-based methods is their inherent explainability. Decisions are based on explicit rules or knowledge, making it easier to understand and interpret the reasoning behind the system's outputs.

However, it's essential to acknowledge the limitations of knowledge-based methods. While effective in scenarios with explicit rules, they may face challenges in handling uncertainty or adapting to dynamic and complex environments.

In summary, knowledge-based methods offer a transparent and rule-driven approach to decision-making, and their collaboration with data-driven approaches in hybrid systems showcases their adaptability and relevance in diverse applications.

## 2.6 Hybrid Approaches

Hybrid approaches in the field of AI represent a beneficial integration of both knowledge-based and data-driven methods. These approaches utilise the strengths of each method to address the limitations inherent in independent systems, fostering a more comprehensive and adaptable decision-making framework.

Knowledge-based methods rely on explicit rules, ontologies, and expert systems to encode human expertise and domain-specific information. While effective in well-defined environments, these approaches may struggle with handling uncertainty and evolving data patterns. On the other hand, data-driven methods, such as ML algorithms, excel in learning patterns and making predictions from large datasets but may lack interpretability and struggle in domains with limited labelled data.

In the research conducted by Wilhelm et al. [2], three fundamental strategies for integrating hybrid approaches have been identified, illustrated in Figure 2.4. These strategies provide valuable insights into how different facets of knowledge-based and data-driven methods can be effectively combined to enhance the overall performance of AI systems.

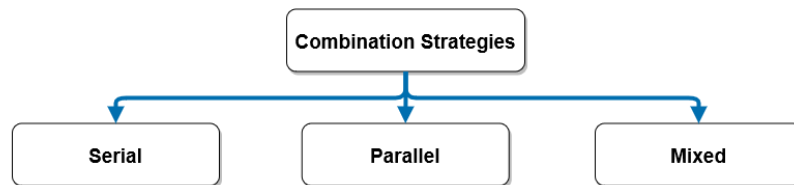


Figure 2.4: Combination Strategies for Hybrid Approaches [2].

In serial combination strategies, multiple fault detection and diagnosis methods are sequentially integrated, where the output of one method serves as the input for the next. This sequential fusion allows for the incorporation of diverse approaches at different stages of the process, from acquiring process measurements to decision-making. By selecting the most suitable method for each transformation step, this approach enhances flexibility. However, compatibility between the interfaces of consecutive methods is crucial [2].

Parallel combination strategies involve merging the results of multiple fault detection methods executed independently and concurrently. Each method addresses the problem autonomously, formulating its decision-making strategy without reliance on other methods. This approach enhances stability and reliability by introducing method redundancy. The redundancy helps mitigate errors or inaccuracies in individual methods, thereby improving decision-making in uncertain conditions [2].

Finally, the integration of serial and parallel fusion strategies can enhance the overall performance of fault detection systems, allowing for a combined fusion of methods, decisions, and a priori knowledge. This amalgamation offers improvements in fault detection by utilising parallel methods based on data, generating complementary sets of residuals. Further enhancements involve serial combinations for fault symptom generation using threshold functions, which can then be leveraged by a data-driven classifier for precise fault determination [2].

Knowledge integration plays a pivotal role across various categories of the entire process flow, influencing critical stages including pre-processing, model training, and model design, as emphasised by Gay et al. in their study [24].

In specific scenarios, the integration of expert knowledge may not be immediately apparent, as noted by Gay et al. [24]. An example can be found in the work of Diez-Olivan et al. [25], where the learning framework is executed through a machine-learning approach. This approach employs a constrained K-Means clustering for outlier detection with fuzzy modelling of distances to normality. In such instances, the subtle integration of expert knowledge may be deeply incorporated within the ML methodology, contributing to the system's robustness and performance.

In the pre-processing phase, knowledge incorporation involves using domain expertise to enhance data cleansing, normalisation, and feature engineering. Experts can provide valuable insights to refine data handling procedures, ensuring that the input data is appropriately conditioned for subsequent stages. This proactive involvement of knowledge at the pre-processing stage sets a strong foundation for downstream tasks. This type of knowledge frequently manifests in the form of physics-based equations that establish connections between input data, as highlighted by Gay et al. in their study [24].

Daw et al. [26] introduced a framework that capitalises on the integration of knowledge from physics-based models with a data-driven model (Neural Network), resulting in improved overall accuracy and robustness. In [16], domain knowledge is utilised to categorise and streamline the initially recorded 44 turbine states into four distinct categories. This categorisation occurs before the application of data-mining algorithms, aiming to reduce the computational effort required and enhance the efficiency of subsequent analyses.

As previously mentioned, the work by Hu et al. [12] involves incorporating additional features into the dataset obtained from a wind turbine. These extra features are derived from the existing operational data using domain-specific knowledge.

Furthermore, expertise plays a crucial role in shaping how models are designed. It influences the choices made about the model's structure, the layers included, and the overall design. Experts in the field can provide essential insights to customise the model to fit the details of the specific problem, making it work better and easier to understand.

In [27], a refined Neural Network architecture was introduced, incorporating fuzzy numbers to capture and incorporate knowledge. In this modified architecture, traditional neurons are substituted with fuzzy equivalents, using fuzzy inputs, outputs, and weights.

Fuzzy numbers, integral to fuzzy logic, are numerical representations that introduce a degree of uncertainty or imprecision into data. Defined by membership functions, these numbers express the extent to which specific values belong to the fuzzy set.

In [28], a rule-based expert system was utilised to apply a set of rules derived from fuzzy data, serving as both inputs and outputs for training a Neural Network. Typically, these rules are expressed in the form: IF condition, THEN consequence. Furthermore, In the study by [29], domain knowledge was used to handle the characteristics of data, allowing for the customisation of a Genetic Algorithm employed in feature extraction.

The learning phase is crucial, involving the adjustment of model weights to align with the available dataset. Particularly in ML, including Neural Networks, this step requires careful attention. Learning optimisation can be achieved by incorporating knowledge into the loss function. In [26], domain knowledge encapsulated in equations between target and input variables integrated these relationships into the loss function as penalties for violations in a Neural Network.

A summary of the reviewed papers is reported in Table 2.3.

Table 2.3: Summary of the reviewed papers.

| Integration Stage | References | Description                                                                                                                                                                                 |
|-------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Pre-processing    | [26]       | Combination of knowledge from physics-based models with a Neural Network.                                                                                                                   |
|                   | [16]       | Domain knowledge is applied to categorise 44 turbine states into four categories before applying data-mining algorithms. This enhances computational efficiency and analysis effectiveness. |
|                   | [12]       | Incorporation of additional features into the dataset by utilising domain-specific knowledge.                                                                                               |
| Model Design      | [27]       | Introduction of a refined Neural Network architecture incorporating fuzzy numbers to capture and incorporate knowledge.                                                                     |
|                   | [28]       | Utilisation of a Rule-Based Expert System to apply a set of rules derived from fuzzy data                                                                                                   |
|                   | [29]       | Utilisation of domain knowledge to handle the characteristics of data, allowing for the customisation of a Genetic Algorithm employed in feature extraction.                                |
| Fine Tuning       | [26]       | Incorporation of domain knowledge through equations between target and input variables into the loss function as penalty terms in a Neural Network.                                         |

## 2.7 Performance Metrics

Performance metrics are crucial tools used to assess the performance of ML models and algorithms. These metrics provide quantitative measures that help to understand how well a model is performing on a given task. The choice of evaluation metrics depends on the nature of the problem being solved and the goals of the analysis.

One fundamental metric for failure prediction classification tasks is accuracy, which measures the ratio of correctly predicted instances to the total number of instances. While accuracy is a common and intuitive metric, it may not be suitable for imbalanced datasets, where one class

significantly outnumbers the others. In such cases, precision, recall, and F1 Score become valuable alternatives.

Precision measures the proportion of correctly predicted positive instances among all failure predictions, emphasising the model's ability to avoid false failure predictions. On the other hand, recall quantifies the model's capacity to identify all positive instances, including the ones it may miss. The F1 Score combines precision and recall into a single metric. This balanced measure is particularly useful in evaluating binary classification models, especially when there is an uneven distribution between the classes. The area under the curve metric is commonly used in binary classification problems. It quantifies the model's ability to distinguish between positive and negative instances.

As per the findings reported in [5], Table 2.4 summarises the utilisation percentages of evaluation metrics across 34 studies focused on failure prediction in industrial maintenance.

Table 2.4: Performance metrics utilised in the papers reviewed in [5].

| Performance Metric | Percentage of Use |
|--------------------|-------------------|
| Recall             | 71%               |
| Precision          | 59%               |
| Accuracy           | 47%               |
| F-score            | 32%               |
| AUC                | 18%               |

## 2.8 Model Explainability

AI and ML models are increasingly being integrated into various industrial settings. These models have demonstrated remarkable capabilities in enhancing predictive accuracy, optimising operations, and driving innovation. However, despite their impressive performance, these advanced models present significant challenges regarding transparency, accountability, and trust [30].

The complexity inherent in modern AI and ML models poses a barrier to understanding their decision-making processes [31]. This opacity can raise concerns among engineers, operators, and decision-makers who rely on these models for critical decisions. Ensuring transparency is crucial to building trust among users, enabling them to understand and validate the model's predictions. Therefore, the demand for XAI is growing, aiming to make AI models more interpretable, understandable, and trustworthy without compromising performance [30].

### 2.8.1 Explainable Artificial Intelligence

XAI has emerged as an essential area of AI aimed at addressing these challenges by making AI models more interpretable and understandable to humans. The primary goal of XAI is to make the decision-making processes of AI models clearer and more comprehensible to humans, thereby enhancing their transparency and trustworthiness. This is particularly important in critical areas where understanding why a model made a certain prediction is crucial for ensuring safety, regulatory compliance, and ethical AI deployment.

In industrial settings, engineers often require insights into why a piece of machinery is predicted to fail to undertake appropriate corrective actions and prevent downtime. The explainability of failure prediction models becomes imperative for several reasons:

**Trust and Adoption:** Enables users to trust and adopt models whose predictions they can understand. Transparent models facilitate better acceptance and integration into decision-making processes.

**Improved Decision-Making:** Understanding the factors leading to a predicted failure enables more effective and targeted interventions, ultimately leading to better outcomes.

**Model Validation:** Explainable models empower developers and users to identify and correct potential biases or errors, enhancing the model's overall reliability and performance.

Model explainability encompasses a range of techniques and approaches designed to make AI models more transparent. These techniques can be broadly categorised into two types, as described in [3] and illustrated in Figure 2.5: intrinsically interpretable models and post-hoc explainability methods.

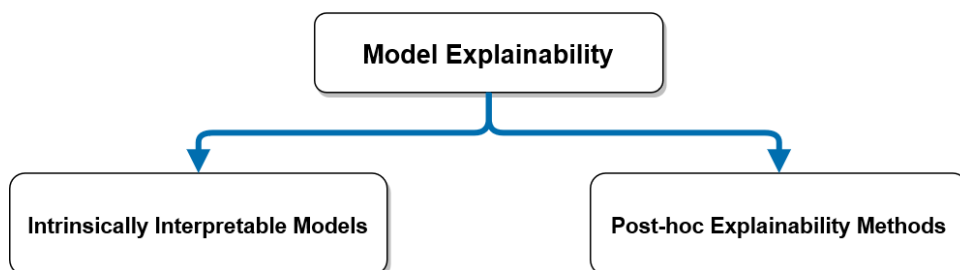


Figure 2.5: Model Explainability techniques [3].

Intrinsically interpretable models are designed in such a way that their decisions can be easily understood by humans. Examples include DTs, rule-based systems, and linear models. These

models have a clear and interpretable structure, making it relatively straightforward to trace how input features influence the model's output.

On the other hand, post-hoc explainability methods aim to provide insights into the workings of complex models after they have been trained. These methods can be further divided into model-agnostic and model-specific techniques.

Model-agnostic methods are designed to be applied to any ML model, regardless of its architecture. These methods operate independently of the model's internal structure, making them highly adaptable and applicable. Examples of model-agnostic methods include:

**SHapley Additive exPlanations (SHAP):** SHAP assigns each feature an importance score indicating its contribution to the model's output. By analysing SHAP values, users can gain insights into which features are driving the model's predictions and how they interact with each other.

**Local Interpretable Model-agnostic Explanations (LIME):** LIME approximates the model's behaviour locally around a specific instance by training an interpretable surrogate model. LIME provides insights into how the model's decision would change if the input features were modified slightly, helping users understand the model's decision-making process. This method is particularly useful for generating explanations for individual predictions.

Model-specific methods are tailored to specific types of models and exploit their unique properties to provide explanations. These methods often provide more precise and detailed insights because they can exploit the internal structure of the model they are designed for. Examples of model-specific methods include:

**Gradient-weighted Class Activation Mapping (Grad-CAM):** Grad-CAM is used in computer vision tasks to generate visual explanations for convolutional neural networks. It highlights regions of input images that are most relevant to the model's predictions, allowing users to understand which parts of the image influenced the model's decision. Grad-CAM works by using the gradients of the target class flowing into the final convolutional layer to produce a coarse localisation map [3].

**Feature importance scores in tree-based models:** DTs are inherently transparent and have been widely used in decision-making contexts due to their simplicity and ease of interpretation. However, as the complexity of DTs increases, their interpretability diminishes. When combined into tree ensembles like RFs to improve predictive performance, they lose their individual transparency [3].

These post-hoc explainability methods, both model-agnostic and model-specific, play a crucial role in enhancing the transparency and interpretability of complex AI models. They enable users to trust and validate their predictions in real-world applications by providing clear and understandable explanations of how models make decisions.

### 2.8.1.1 Local and Global Explainability Approaches

Model explainability can be approached at different levels, primarily categorised into local and global explanations.

Local explainability methods focus on interpreting individual predictions made by a model. These approaches provide insights into why a model made a specific decision for a particular instance, which is crucial for tasks requiring case-by-case analysis.

As previously mentioned, LIME approximates the model's behaviour locally around a specific instance by training an interpretable surrogate model. This approach helps users understand the factors influencing a particular prediction and how slight changes in input features could alter the outcome.

Global explainability methods aim to provide a comprehensive understanding of the overall model behaviour. These approaches elucidate how the model makes decisions across all possible instances, offering insights into general patterns and feature importance.

Partial Dependence Plots illustrate the relationship between a selected feature and the model's predictions, showing how changes in the feature value impact the predicted outcome. This method provides a global view of the feature's effect across the entire dataset, offering valuable insights into the model's behaviour.

### 2.8.1.2 Practical Applications

In [32], both local and global explainability methods were employed to interpret remaining useful life estimations. Explain Like I'm 5 (ELI5)'s global explanations were utilised to analyse the overall influence of each variable on model predictions, complemented by expert knowledge. Additionally, LIME was applied to calculate the contribution of each predictor to the model's output.

In [33], a newly proposed XAI methodology, Failure Diagnosis Explainability (FDE), was integrated into the model to enhance the transparency and interpretability of diagnosed failures. The FDE methodology effectively explained the assessed diagnosis.

Despite its benefits, XAI faces significant challenges. As highlighted in [3], there is a tradeoff between model interpretability and performance, with enhancing interpretability potentially degrading the quality of the model's decisions. This tradeoff poses a critical challenge in deploying XAI in practical applications, where both transparency and predictive accuracy are essential.

Currently, XAI methods are not widely used in the field of failure prediction. While studies such as [32] and [33] demonstrate their potential to enhance interpretability, their adoption in industrial settings remains limited.

## 2.9 Summary and Gap Analysis

This section offers a concise summary of the foundational discoveries supporting this work. An overview of the most significant insights is presented in Table 2.5. While data-driven methods are extensively employed, there remains a notable gap in the adoption of hybrid approaches that integrate data-driven methods with knowledge-based methods. This dissertation aims to solidify the utilisation of knowledge-based methods by incorporating expert knowledge into the predictive maintenance process.

Current predictive maintenance methodologies have notable gaps. While data-driven approaches are prevalent, they often overlook the valuable insights domain experts provide. Moreover, the lack of transparency in ML models poses a significant challenge, hindering trust and interpretability. Additionally, many existing systems lack mechanisms for continuous improvement based on real-time feedback, resulting in models that struggle to adapt to changing operational environments. These gaps highlight the need for innovative hybrid solutions that integrate data-driven methods with domain expertise while also prioritising transparency and adaptability.

This dissertation not only addresses a crucial gap in the existing landscape but also aspires to position the research at the forefront of advancing predictive maintenance methodologies. By integrating the strengths of both data-driven and knowledge-driven approaches, this study aims to achieve a more robust predictive maintenance system. The incorporation of expert knowledge promises not only to enhance accuracy but also to provide a more comprehensive understanding of equipment conditions.

Table 2.5: Overview of the related research.

| Reference                                                                                                                                    | Approach                 | Findings                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Li et al. [8]<br>Canizo et al. [9]<br>Leahy et al. [10]<br>Susto et al. [11]<br>Hu et al. [12]                                               | Pre-Processing           | PCA stands out as the most widely employed method for dimensionality reduction. Additionally, techniques including the elimination of constant variables, utilising domain knowledge, and handling missing values, have also found application in this context.                                                                                                                                                                                        |
| Li et al. [8]<br>Kusiak et al. [16]<br>Chen et al. [17]<br>Orrù et al. [18]<br>Alves et al. [19]<br>Xiang et al. [20]<br>Binding et al. [21] | Supervised ML Algorithms | RF consistently demonstrated high accuracy across different studies, making it a reliable choice for various predictive maintenance tasks.                                                                                                                                                                                                                                                                                                             |
| Sun et al. [22]<br>Chi et al. [23]                                                                                                           | Knowledge-Driven Methods | Rule-Based Systems, Expert Systems and Ontologies, play a crucial role in decision-making by incorporating explicit rules and expert knowledge. Despite their inherent explainability, it is crucial to recognise the limitations of knowledge-based methods, especially in handling uncertainty and adapting to dynamic environments.                                                                                                                 |
| Hu et al. [12]<br>Kusiak et al. [16]<br>Gay et al. [24]<br>Daw et al. [26]<br>Satish et al. [27]<br>Garga et al. [28]<br>Li et al. [29]      | Hybrid Approaches        | Expertise was shown to be essential in the pre-processing phase, influencing data cleansing, normalisation, and feature engineering. In certain scenarios, the integration of expert knowledge may not be explicitly evident but can significantly contribute to the robustness and performance of ML methodologies. Adopting a hybrid approach that incorporates domain knowledge at different stages of ML processes can bring substantial benefits. |
| Minh et al. [30]<br>Linardatos et al. [31]<br>Barredo Arrieta et al. [3]<br>Serradilla et al. [32]<br>Zeldam [33]                            | XAI                      | AI in industry enhances accuracy but lacks transparency. Methods like LIME and SHAP provide local explainability, while PDP offers global insights. Industrial adoption of XAI methods remains limited.                                                                                                                                                                                                                                                |



## Chapter 3

# Implementation

Chapter 3 explores the methodology employed in this project, outlining the steps taken to develop and implement the system. Firstly, Section 3.1 introduces the proposed system architecture. Following that, Section 3.2 discusses the dataset used in this study. Section 3.3 examines the data pre-processing steps applied. Section 3.4 details the implementation of the failure prediction model, encompassing both the data-driven and knowledge-driven approaches. Section 3.5 covers the integration of XAI methods. Section 3.6 presents the implementation of the web application developed. Finally, Section 3.7 provides a summary of the Chapter.

### 3.1 System Architecture

The development of the failure prediction and reporting system involved several critical stages: data reception and pre-processing, model development, rule-based adjustments, explainability through XAI, user-implemented rules, and failure reporting with model retraining, as illustrated in Figure 3.1.

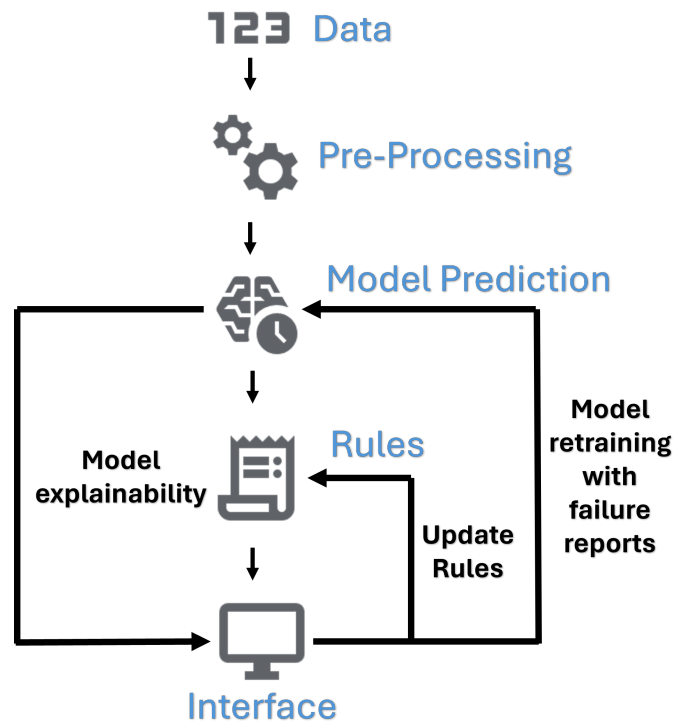


Figure 3.1: Proposed system architecture.

The system architecture comprises several stages designed to ensure the quality and relevance of the input data, develop accurate predictive models, and maintain system transparency and user engagement:

- **Data Pre-processing:** This step involved data cleaning, dimensionality reduction, correlation assessment, and feature engineering to prepare the data for model training.
- **Failure Prediction Model Development:** ML algorithms were applied to predict failures. These predictions were further refined by incorporating expert-derived rules, enhancing accuracy and robustness in identifying nuanced failure patterns.
- **Integrating Model Explainability:** To enhance system transparency and empower users to implement their own rules, an XAI method was employed, providing insights into the model's decision-making process.
- **Web Application Development:** All tasks were integrated into a web application, which facilitates the visualisation of results, the integration of new rules, and the submission of failure reports that trigger the retraining of the ML model.
- **System Evaluation and Validation:** The system's predictions were evaluated and validated using metrics such as F1 Score, recall, and precision.

The system was developed using Python to guarantee flexibility. Each stage was designed to ensure robust, accurate, and transparent predictions, promoting continuous improvement and user engagement. The integration of various technologies and a validation strategy contributed to the system's reliability.

## 3.2 Dataset

The primary objective of this project is to address and predict failures within the metro system's air production units. Failures in these units can lead to significant operational disruptions, impacting the reliability and efficiency of metro services. By analysing the dataset, it aims to identify patterns and anomalies that precede equipment failures. This predictive maintenance approach can help in preemptively addressing issues before they escalate into major faults, ensuring smooth and uninterrupted metro operations.

For the purpose of testing and implementing this project, a dataset from a metro train, similar to the one exemplified in Figure 3.2, in an operational context, operated by "Metro do Porto", was utilised [34]. This publicly available dataset includes fifteen features such as pressure readings, temperature, motor current, and air intake valve data collected from a compressor's Air Production Unit between February and August 2020. The data was logged by an onboard embedded device, resulting in a total of 1,516,948 instances.



Figure 3.2: Image of a "Metro do Porto" metro train (retrieved from [4]).

Initially unlabeled, the dataset underwent a labelling process using failure reports provided by the metro company, as exemplified in Table 3.1. This process involved correlating sensor data with documented failure events to create a labelled dataset. A new feature, "Failure", was introduced where "1" indicates a failure and "0" denotes no failure. Given that failure reports were compiled by experts from April to July 2020, only data within this timeframe was used to train and test the subsequently developed predictive model.

Table 3.1: Failure reports provided by field-experts [6].

| Nr. | Start Time      | End Time        | Duration (min) |
|-----|-----------------|-----------------|----------------|
| 1   | 4/12/2020 11:50 | 4/12/2020 23:30 | 700            |
| 2   | 4/18/2020 00:00 | 4/18/2020 23:59 | 1440           |
| 3   | 4/19/2020 00:00 | 4/19/2020 01:30 | 90             |
| 4   | 4/29/2020 03:20 | 4/29/2020 04:00 | 40             |
| 5   | 4/29/2020 22:00 | 4/29/2020 22:20 | 20             |
| 6   | 5/13/2020 14:00 | 5/13/2020 23:59 | 599            |
| 7   | 5/18/2020 05:00 | 5/18/2020 05:30 | 30             |
| 8   | 5/19/2020 10:10 | 5/19/2020 11:00 | 50             |
| 9   | 5/19/2020 22:10 | 5/19/2020 23:59 | 109            |
| 10  | 5/20/2020 00:00 | 5/20/2020 20:00 | 1200           |
| 11  | 5/23/2020 09:50 | 5/23/2020 10:10 | 20             |
| 12  | 5/29/2020 23:30 | 5/29/2020 23:59 | 29             |
| 13  | 5/30/2020 00:00 | 5/30/2020 06:00 | 360            |
| 14  | 6/01/2020 15:00 | 6/01/2020 15:40 | 40             |
| 15  | 6/03/2020 10:00 | 6/03/2020 11:00 | 60             |
| 16  | 6/05/2020 10:00 | 6/05/2020 23:59 | 839            |
| 17  | 6/06/2020 00:00 | 6/06/2020 23:59 | 1439           |
| 18  | 6/07/2020 00:00 | 6/07/2020 14:30 | 870            |
| 19  | 7/08/2020 17:30 | 7/08/2020 19:00 | 90             |
| 20  | 7/15/2020 14:30 | 7/15/2020 19:00 | 270            |
| 21  | 7/17/2020 04:30 | 7/17/2020 05:30 | 60             |

Experts have indicated that two hours are necessary to repair the issue in a timely manner. To provide operators with enough time to address issues before an actual failure occurs, labels indicating a failure were assigned starting two hours before the reported failure time. This preemptive labelling strategy aimed to provide a sufficient lead time for operators to intervene and potentially prevent failure, thereby enhancing the system's effectiveness in facilitating proactive maintenance.

The following Section will detail the analysis and pre-processing conducted on this data.

### 3.3 Data Pre-processing

Pre-processing is a critical step in data preparation for data-driven models, significantly influencing their accuracy and performance. The implemented steps include feature engineering, correlation assessment, and dimensionality reduction, which will be detailed in this Section. Since the dataset had no missing values, there was no need for imputation or removal steps related to missing or inconsistent data.

Feature engineering involved extracting important insights from raw sensor data to enhance predictive capabilities. An analysis of various aggregation intervals from 5 minutes to 30 minutes determined that a 15-minute interval best balanced capturing dynamics while smoothing data noise. Once the aggregation interval was determined, the next step involved computing aggregate statistics, with the median pressure value identified as a crucial metric for capturing the central tendency of the data within each interval. The median was preferred over other measures, such as the mean or maximum, to reduce the influence of outliers and extreme values, which could distort the aggregated results. These aggregated features replaced original ones, simplifying the dataset for improved model performance.

The impact of employing this feature engineering approach became apparent during the subsequent model training and evaluation phases. Models trained on these engineered features consistently outperformed those trained solely on raw sensor data. This improvement highlighted enhanced predictive accuracy and robustness.

Following that, a feature autocorrelation analysis was performed. The main goal of this step was to reduce the number of feature columns by identifying irrelevant variables, such as variables with no correlation to failures and redundant ones.

Consequently, the correlation coefficients between the target variable, "Failure", and all other features in the dataset were calculated using the Pearson method. To filter out variables that showed no significant impact on system failure prediction, a correlation threshold of 0.21 was applied. Features with correlation coefficients below this threshold were considered less relevant and were subsequently removed from the dataset to optimise the model's training process, resulting in the removal of four variables. Following that, the pairs of features with high correlation coefficients were identified, considering a threshold of 0.81. One feature from each pair exhibiting high correlations with others was removed to avoid redundancy, resulting in four feature removal.

To facilitate this assessment and mitigate multicollinearity issues, a correlation matrix of the features was visualised using a heatmap, depicted in Figure 3.3.

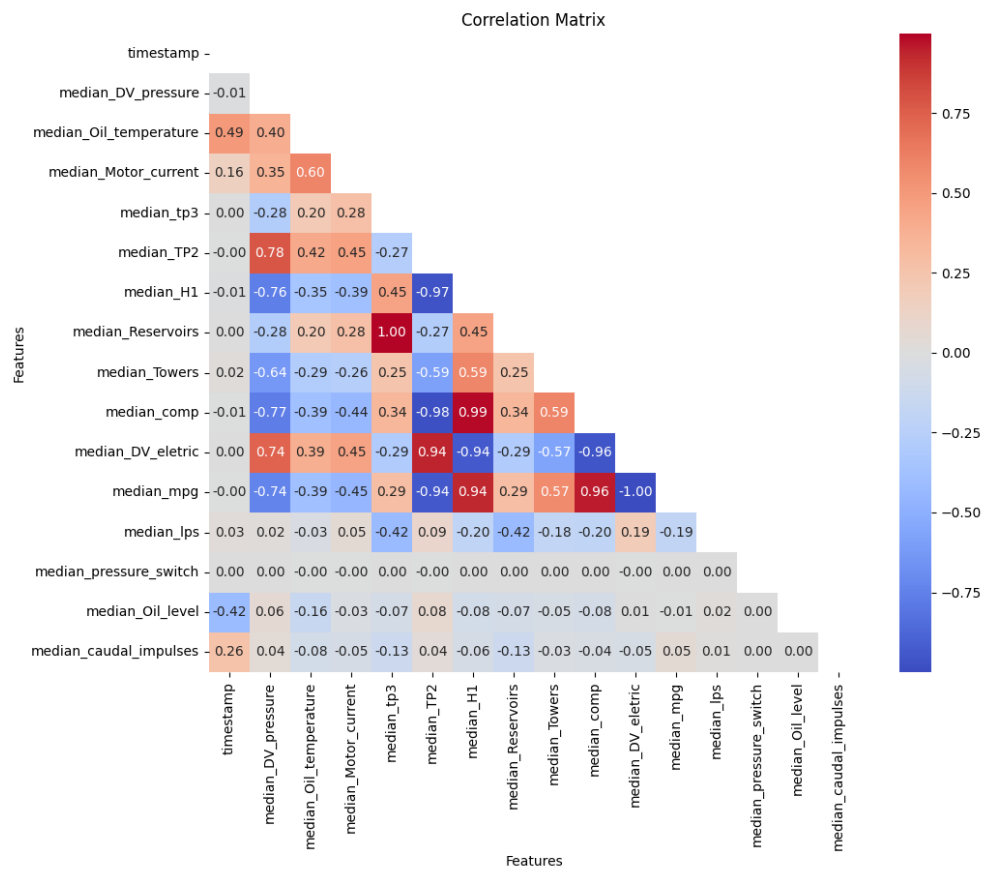


Figure 3.3: Correlation matrix calculated using Pearson correlation coefficient.

Furthermore, PCA, implemented with the Scikit-learn library, was explored to reduce dataset dimensionality and capture significant data variance. This process began by selecting the number of components for the PCA analysis. These components represent the dimensions that capture the most significant variance in the data. For that, a scree plot was generated, which displays the explained variance for each principal component, as can be observed in Figure 3.4.

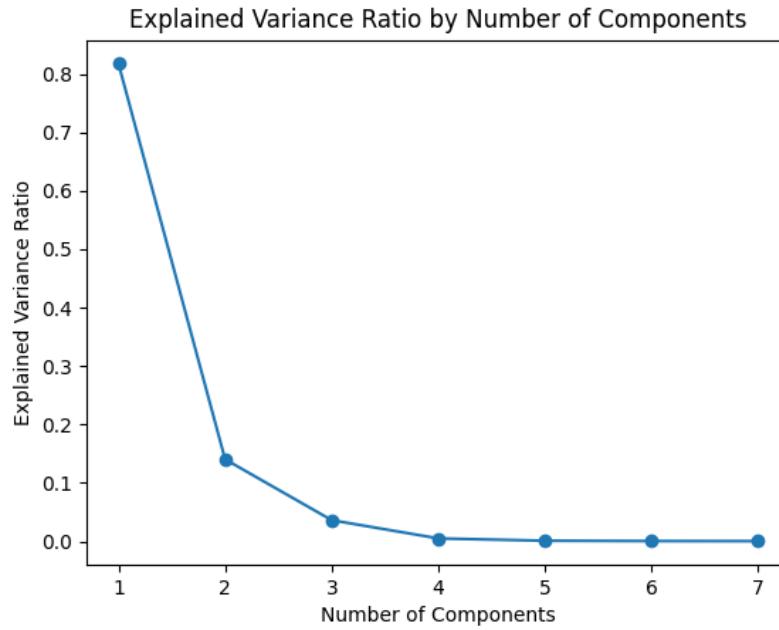


Figure 3.4: Scree plot showing the explained variance ratio by number of components in PCA.

The "elbow" point in the scree plot indicates the optimal number of components. This observed optimal number was the one considered during the PCA implementation. However, applying PCA for dimensionality reduction led to a decline in model performance. This outcome is likely due to the loss of critical information essential for failure prediction, given the already low dimensionality of the original feature set. Consequently, PCA was not integrated into the final model.

Finally, the dataset was temporally split into training and testing sets, with the testing set comprising 417,522 samples from June 4, 2020, onward. These pre-processing steps collectively enhanced the dataset's suitability for training a more accurate data-driven model for failure prediction.

## 3.4 Failure Prediction Model

This Section presents the implementation of the failure prediction model, which integrates a data-driven approach combined with knowledge-based methods. This hybrid methodology ensures robustness and enhances the accuracy of failure predictions. Subsection 3.4.1 details the implementation of the data-driven approach, and subsection 3.4.2 presents the knowledge-driven method.

### 3.4.1 Data-driven Approach

Although an extensive amount of data was available, it only holds value if significant information can be extracted from it. Therefore, ML techniques were employed to extract meaningful insights from it.

The dataset, now labelled, facilitated the implementation of supervised learning ML methods, focusing primarily on classification tasks. Various algorithms were considered, including RF, XGB, CatBoost, GBM, LightGBM, and a voting ensemble. RF was chosen based on its widely recognised efficacy and performance in the literature. Similarly, boosting algorithms like XGB, CatBoost, and GBM were selected for their proven success in predictive accuracy. LightGBM was included for its efficiency and lightweight nature. Although the literature also indicated that the SVM model is effective for failure prediction, early development tests with an SVM model produced poor results across all metrics and required extensive training time, leading to its exclusion from the final selection.

A voting ensemble algorithm was utilised to consolidate predictions from multiple models, enhancing overall prediction accuracy and helping to prevent model overfitting. This approach combines outputs from diverse models to enhance overall prediction accuracy. Specifically, XGB, CatBoost, GBM, and LightGBM were selected for their demonstrated strong performance. Each model's contribution to the ensemble is weighted based on its recall value, which was normalised to ensure that models with higher recall influenced the final prediction more significantly.

The ML algorithms were implemented using Python libraries such as Scikit-learn, XGB, LightGBM, and CatBoost.

Given the dataset's inherent class imbalance, where failure events are relatively infrequent compared to normal operations, evaluating these models using appropriate metrics was critical. Performance assessment focused on recall, precision, and F1 Score metrics. Among these, the primary focus was placed on the recall metric due to its importance in identifying the maximum number of failure events, thereby minimising false negatives and ensuring robust, proactive maintenance strategies.

The models were trained on the previously cleaned and processed data. Hyperparameter tuning, utilizing the Grid Search method from Scikit-learn, was employed to optimise model performance. This approach ensured that the best combination of hyperparameters was selected for each model, enhancing predictive capabilities by fine-tuning parameters like the number of trees in RF, learning rate in XGB and CatBoost, and boosting iterations in GBM and LightGBM to achieve optimal metric scores.

### **3.4.2 Knowledge-driven Approach**

To enhance the data-driven model's predictions, a set of rules was defined based on domain knowledge and insights gained from an Exploratory Data Analysis (EDA). The choice of a rule-based system over other knowledge-based methods was driven by its straightforward implementation and ability to handle specific, nuanced scenarios. Rules allow for precise adjustments and immediate application of expert insights, which is crucial for improving accuracy, addressing edge cases, and ensuring that rare but critical scenarios are correctly managed. This approach provides a flexible and practical way to incorporate domain expertise directly into the predictive model, enhancing its reliability and effectiveness in real-world applications.

The EDA was conducted to gain insights into the dataset and identify patterns that could inform rule creation. The analysis began by generating descriptive statistics and summarising key metrics such as mean, standard deviation, and percentiles for each feature. This provided a comprehensive overview of the data characteristics, identifying anomalies and outliers that could influence model performance.

Following this, time series visualisations, as the ones illustrated in Figure 3.5, were employed to observe temporal trends and fluctuations. This type of visualisation was considered for its ability to offer a dynamic view of how each feature evolves across temporal intervals, offering insights into their patterns and variations. This detailed exploration revealed potential correlations and periods of high variability, essential for understanding the factors influencing failure prediction and optimising the system performance.



Figure 3.5: Time series plots generated during the Exploratory Data Analysis.

Kernel Density Estimation (KDE) plots were generated to visualise feature probability densities, as exemplified in Figure 3.6. These plots provide a smoothed estimate of the data distribution, allowing a clearer understanding of the underlying structure and variability within each feature. This technique helped identify potential outliers and overall data density patterns that may influence model performance and analytical decisions.



Figure 3.6: Kernel Density Estimation plots illustrating data distribution characteristics, generated during the Exploratory Data Analysis.

The domain knowledge derived from EDA was proven crucial in refining the rule-based adjustments within the failure prediction system. EDA helped to reveal important data patterns, particularly the correlation between high oil temperatures and failure occurrences, as well as the combined influence of oil temperature and differential pressure on equipment failure risk. These insights guided the formulation of the following specific rules for more precise adjustments to the model's predictions:

- **Rule 1 - High Oil Temperature:** Instances of oil temperatures exceeding 83°C were rare but often coincided with equipment failures, as revealed by statistical and KDE analysis. This insight led to Rule 1, which adjusts the model's predictions to signal a failure when

such high temperatures are observed. If the initial prediction is 0 (no failure) but the median oil temperature exceeds 83°C, the prediction is adjusted to 1 (failure).

- **Rule 2 - Low Oil Temperature:** Temperatures below 67.25°C typically indicated normal operating conditions with minimal failure risk. Rule 2 was created to correct predictions that wrongly flagged failures when oil temperatures fell below this threshold. If the initial prediction is 1 (failure) but the median oil temperature is below 67.25°C, the prediction is adjusted to 0 (no failure).
- **Rule 3 - Combined Oil Temperature and Differential Pressure:** Statistical exploration highlighted significant relationships between oil temperature and differential pressure readings. Specific combinations, such as high oil temperatures and low differential pressure, were indicative of imminent equipment failures. Rule 3 utilised this insight to adjust predictions based on critical combinations of these factors. If the initial prediction is 0 (no failure) and both the median oil temperature exceeds 75.65°C, and the median differential pressure exceeds -0.02, the prediction is adjusted to 1 (failure).

These rules fine-tune the data-driven model's predictions following its initial assessment, thereby enhancing accuracy through the integration of crucial domain insights. Designed to optimise predictive outcomes, these rules ensure the model responds appropriately to conditions indicative of potential failures.

### 3.5 Explainability Incorporation

To enhance system transparency and enable continuous improvement through rule implementation, the XAI method LIME was integrated. LIME was chosen for its interpretable rules, enabling domain experts to understand how predictions are made and facilitating their contribution to rule refinement. The implementation of LIME using the Lime Python library involved training the explainer with available data to capture underlying feature distributions and relationships. Specifically, LIME's "classification" mode was employed to generate textual explanations for individual predictions, revealing the specific conditions or rules influencing each outcome. This textual format was chosen over graphical plots to provide detailed insights into how each feature contributes to the model's predictions.

This approach empowers experts to identify and prioritise critical factors impacting the model's decisions, thereby improving its accuracy and reliability over time. By exploiting and presenting LIME's results, the system achieves interpretability, enabling users to gain new insights into the underlying process and deepen their understanding of the system's behaviour. This transparency not only supports better decision-making but also facilitates the refinement and optimisation of the failure prediction system.

## 3.6 Application

A web application was developed to serve as a real-time monitoring tool, enabling the visualisation of results from the implemented failure prediction and reporting system. The application and its graphical interface were built using Flask, a Python web development framework. It integrates the steps and system processes described in earlier sections, including sensor data processing, failure prediction using a data-driven ML model enhanced with knowledge-driven insights in the form of rules, and the visualisation of explainability methods results.

To handle messages and enable communication between the sensors and the web application, the Message Queuing Telemetry Transport (MQTT) protocol was used. Additionally, the web application is connected to a PostgreSQL database.

MQTT is a lightweight messaging protocol that operates on a publish-subscribe model where devices publish messages to topics, serving as channels for data transmission. A MQTT subscriber was implemented to receive the data in the web application. Once the application receives an MQTT message, the following steps are executed for failure prediction:

- **Message Decoding and Extraction:** The received message is decoded to extract the sensor data, which includes multiple feature values and a timestamp.
- **Storing Data in a Rolling Window:** The extracted data is parsed and stored in a deque structure. This deque maintains a rolling window of the most recent 900 data points. Given that messages are received every 10 seconds, this corresponds to 15 minutes of data. The rolling window ensures the application has sufficient data for feature engineering.
- **Feature Engineering:** The raw data is processed to compute median values for selected features over 15-minute intervals.
- **Prediction:** Using the processed sample, the pre-trained data-driven prediction model predicts potential failures.
- **Rule Application:** The system evaluates the predictions against predefined rules. These rules help refine the predictions by applying additional conditions based on the feature values.
- **Explanation Generation:** To provide insights into the predictions, LIME generates explanations that highlight the impact of each feature on the prediction. These explanations are presented in a clear, text-based format, making it easier for users to understand the factors influencing the predictions.
- **Database Storage:** The application enhances database write efficiency by batching the processed data, including feature values, timestamps, and predictions, into groups of 10 operations before storing them in the database. This batching strategy reduces the frequency of database writes, improving performance while maintaining data integrity.

Six application pages were implemented, with their interface developed using a combination of HTML, JavaScript, and CSS to provide a responsive experience. HTML was used to structure the content of the web pages, JavaScript was employed to implement interactive features and dynamic content updates, and CSS was used to style and ensure a consistent visual design across the application.

User interface (UI) mockups were created prior to implementation to facilitate thorough planning. These mockups, developed using Figma, are presented in Figures 3.7, 3.8, 3.9, 3.10, 3.11, and 3.12. In the following subsections, the functionalities and pipelines triggered by each of the pages in the system will be detailed.

### 3.6.1 "Dashboard" Page

The "Dashboard" page (Figure 3.7) displays the results of failure predictions based on incoming data, allowing for immediate user intervention. When a failure is predicted, a failure message appears in red, providing a clear visual cue to alert users to the critical situation. This immediate visual indicator ensures that potential issues are quickly recognised and addressed.

In addition to failure predictions, real-time values from various sensors, such as pressure, temperature, motor current, and air intake valve readings, are displayed. This gives users a comprehensive view of the operational status and helps to immediately diagnose potential issues.

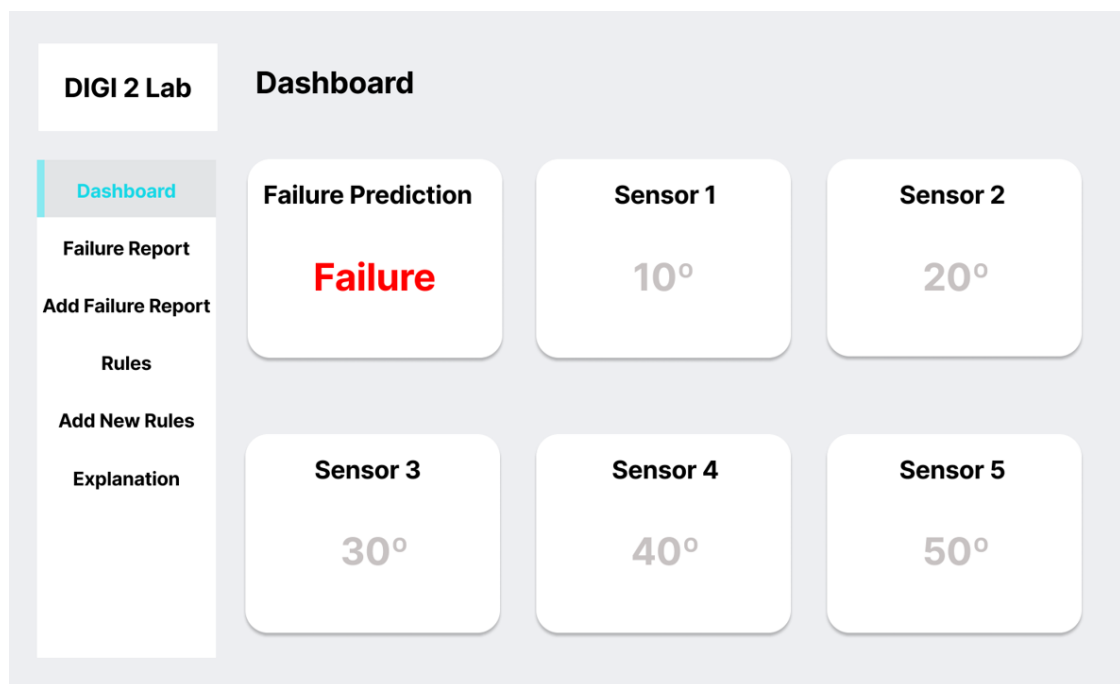


Figure 3.7: UI mockup of the "Dashboard" page.

3.6.2 "Failure Report" Page

The system includes a failure report functionality to maintain accuracy and responsiveness by ensuring continuous learning from new data. Experts can submit reports whenever they identify a failure and access previous ones.

The "Failure Report" page (Figure 3.8) allows users to visualise previously submitted reports and their details.



Figure 3.8: UI mockup of "Failure Report" page.

3.6.3 "Add Failure Report" Page

The "Add Failure Report" page (Figure 3.9) enables users to create new failure reports when identifying a failure. The form includes fields for specifying the start date and time, end date and time, type of failure, and severity. When a failure is observed, experts can report it conveniently through the web interface, providing details about the incident.

Upon submitting a failure report, the system triggers a retraining process for the predictive model. This process begins with updating the database to incorporate the newly reported failure data, ensuring the model’s training dataset reflects the latest insights into failure patterns and conditions.

Next, the retraining process employs incremental learning techniques to enhance the model’s predictive accuracy. By utilising specific conditions within the models that support incremental learning, the model builds upon its existing knowledge without discarding previously learned patterns, focusing exclusively on integrating the new data. This approach allows the model to adapt dynamically to new data, refining its predictions based on the updated dataset.

Integrating this new failure data enables the model to adapt to emerging patterns and enhance its predictive accuracy, ultimately reducing future prediction errors. The continuous feedback loop ensures that the model remains relevant and effective, especially in dynamic industrial environments where conditions and failure modes can change rapidly. By integrating the latest data, the model becomes better equipped to recognise similar patterns and anomalies in the future, enhancing its responsiveness and reliability. The retraining process builds upon existing knowledge and optimises computational resources, enabling effective handling of large datasets and frequent updates. Deploying a retrained model minimises downtime and optimises resource allocation based on predictive insights.

This retraining process ensures that the predictive model evolves with current data, continuously improving its ability to accurately predict failures and optimise operational efficiency in dynamic industrial settings.

The image shows a UI mockup for the 'Add Failure Report' page. On the left is a sidebar with the 'DIGI 2 Lab' logo at the top. Below the logo is a menu with the following items: 'Dashboard', 'Failure Report', 'Add Failure Report' (which is highlighted with a blue background), 'Rules', 'Add New Rules', and 'Explanation'. The main content area on the right is titled 'Add Failure Report'. It contains four dropdown menus, each with a label and a downward arrow icon: 'Start Date:', 'End Date', 'Type of failure:', and 'Severity:'.

Figure 3.9: UI mockup of "Add Failure Report" page.

### 3.6.4 "Rules" Page

As reported in the literature review, knowledge-based methods have limitations in adapting to dynamic environments. Therefore, a system was developed to promote continuous learning and adaptation.

The system encourages domain experts to implement new rules based on their observations and insights, further refining the learning process and accuracy of the predictive model. These insights can then be codified into new rules integrated into the system to refine its predictive accuracy. As experts identify and address new scenarios or anomalies, the rules evolve, making the model more robust over time.

The "Rules" page (Figure 3.10) allows users to manage these prediction rules by viewing existing rules, modifying their priorities, and deleting outdated or incorrect rules to maintain the relevance and accuracy of the rule set. This functionality ensures that users can tailor the predictive model's behaviour to suit specific operational requirements and conditions better.

Users can view the current set of rules that the application applies to predictions. The rules are presented in a user-friendly format, showing the conditions and the corresponding prediction adjustments. Each rule is displayed with its conditions broken down into readable components. For example, a condition like `df['median_Oil_temperature'].iloc[-1] > 83` is displayed as "Oil Temperature > 83", making it easier for humans to understand.

The priority of rules can significantly impact the prediction outcomes. To manage this, users can reorder the rules via a drag-and-drop interface, making it intuitive to change the rule priority. Once the new order is set, the application updates the internal rule list to reflect these changes, ensuring that rules are applied in the specified sequence. If the first condition is met, the subsequent conditions are not checked.

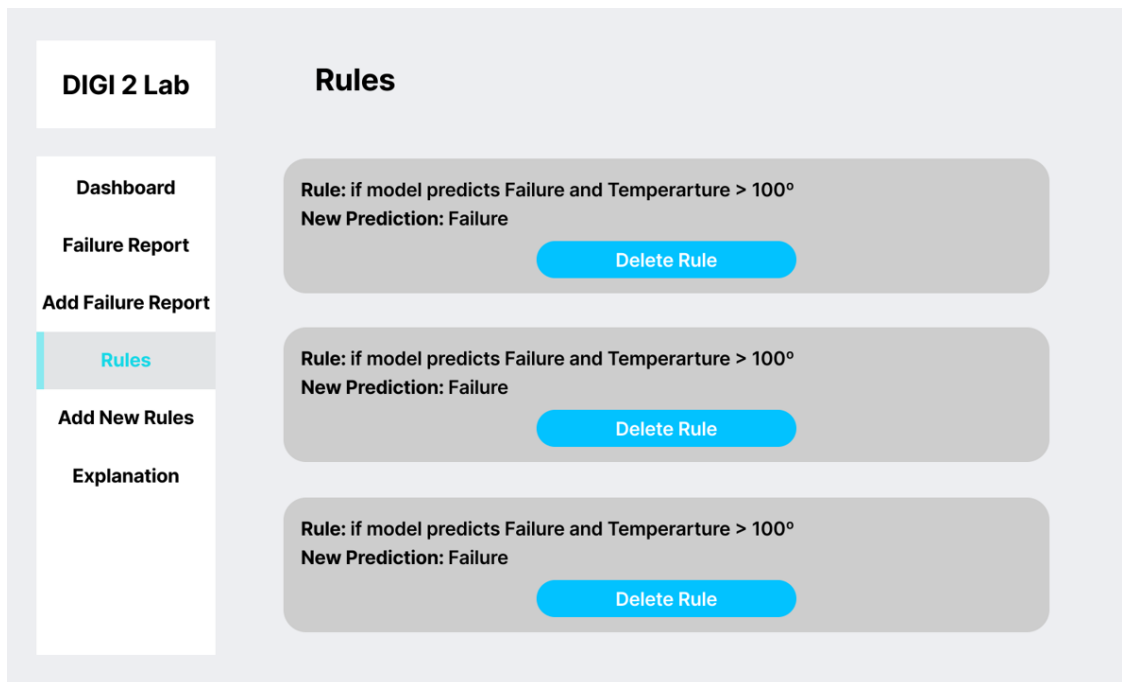


Figure 3.10: UI mockup of "Rules" page.

### 3.6.5 "Add New Rule" Page

On the "Add New Rule" page (Figure 3.11), users can add new rules through a dedicated form. Two types of rules can be defined: single-condition rules and double-condition rules. This flexibility allows users to create complex rules that better reflect the nuances of their operational data.

Users can specify conditions for when the model predicts either "Failure" or "Not Failure" and apply these conditions to various variables using operators and values they choose. The page provides dropdown menus for selecting the prediction outcome, the variable, and the operator,

along with a text input for entering the value. Additionally, users can add more conditions by clicking the "Add more conditions" button.

The image shows a UI mockup for the "Add New Rule" page. On the left is a sidebar with the "DIGI 2 Lab" logo at the top. Below the logo are several navigation links: "Dashboard", "Failure Report", "Add Failure Report", "Rules", "Add New Rules" (which is highlighted with a blue background and white text), and "Explanation". The main content area on the right is titled "Add New Rule". It contains four dropdown menus arranged vertically. The first is labeled "If model predicts:", the second "Select a variable:", the third "Select an operator:", and the fourth "Enter a value:". Each dropdown menu has a white box with a downward-pointing arrow on the right side.

Figure 3.11: UI mockup of "Add New Rule" page.

### 3.6.6 "Model Explanation" Page

The "Model Explanation" page (Figure 3.12) provides detailed explanations of the model's predictions using the LIME XAI method. This method offers insights into the factors influencing each prediction, enhancing transparency and helping users understand the model's decision-making process.

The page displays each feature and its respective score, which indicates the importance or contribution of each feature to the model's prediction. For instance, a positive score for a feature indicates that higher values of that feature are associated with a higher likelihood of machine failure, while a negative score suggests the opposite.

To further aid user understanding, the interface includes an interrogation point next to the score label. When users click on this interrogation point, a pop-up appears with an explanation of what the score means. This pop-up provides detailed information about how the scores are calculated and what they signify, making it easier for users to interpret the model's predictions.

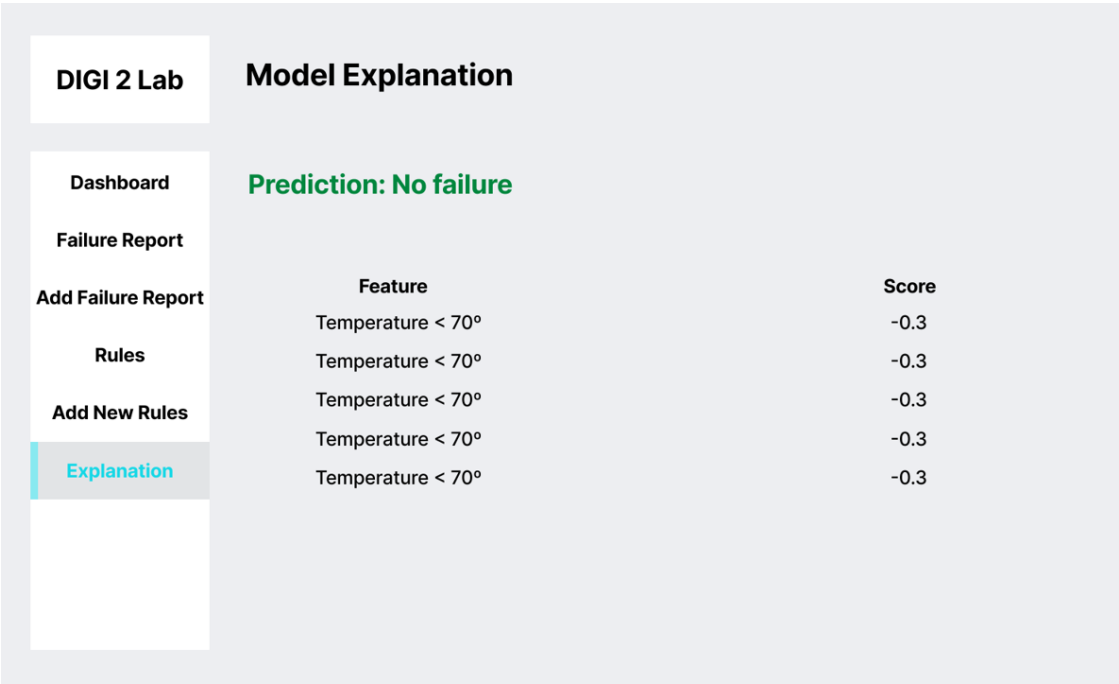


Figure 3.12: UI mockup of "Model Explanation" page.

### 3.7 Summary

Chapter 3 detailed the implementation strategy for developing a real-time monitoring and prediction system aimed at detecting failures. This system integrates data-driven ML techniques with domain knowledge through rule-based adjustments and employs XAI methods to enhance transparency and actionable insights for operators and experts.<sup>1</sup>

Supervised learning algorithms such as RF, XGB, CatBoost, GBM, LightGBM, and a voting ensemble were utilised to extract insights from data and predict potential failures. Hyperparameter optimisation using Grid Search was employed to enhance model performance. A knowledge-driven approach complemented the predictive model by incorporating domain expertise through rule-based adjustments. To improve interpretability, the system integrates XAI methods like LIME, which provide local interpretable explanations for individual predictions. These explanations enable experts to understand the model’s decision-making process and identify areas for refinement.

Additionally, an interactive web application was developed to serve as a real-time monitoring tool. It facilitates the visualisation of results from the implemented failure prediction and reporting system, allowing operators and experts to monitor sensor data in real time, manage prediction rules, submit failure reports, and access detailed model explanations.

<sup>1</sup>The repository for this project is available at: <https://github.com/miguelalmeida8/HyPredictor-Framework>.



## Chapter 4

# Validation and Discussion

Chapter 4 details the performance and results of the failure prediction and reporting system. It begins with an assessment of the data-driven model's performance in Section 4.1. Section 4.2 evaluates the benefits of integrating rule-based adjustments. The discussion on model explainability follows in Section 4.3. The UI results are presented in Section 4.4. Section 4.5 addresses the times taken to compute the systems tasks, and the system validation is covered in Section 4.6. Finally, Section 4.7 provides a summary of the Chapter.

### 4.1 Data-driven Models Performance

The data-driven model's performance was assessed using several metrics: recall, precision, F1 Score, and accuracy. The evaluation results for the six different models - RF, XGB, LightGBM, GBM, CatBoost and the ensemble model - are presented in Table 4.1. This comparison highlights the strengths and weaknesses of each model, allowing for an informed selection based on the specific requirements of the predictive maintenance task.

Table 4.1: Results of the considered evaluation metrics of the implemented Machine Learning models.

| ML Model | Recall (%)    | Precision (%) | F1 Score (%)  | Accuracy (%)  |
|----------|---------------|---------------|---------------|---------------|
| LightGBM | 88.786        | <b>95.767</b> | <b>92.144</b> | <b>99.479</b> |
| CatBoost | 91.413        | 87.288        | 89.303        | 99.247        |
| XGB      | <b>91.991</b> | 88.293        | 90.104        | 99.305        |
| RF       | 90.685        | 50.236        | 64.655        | 96.589        |
| GBM      | 91.003        | 88.193        | 89.576        | 98.886        |
| Voting   | 91.003        | 91.055        | 91.107        | 99.059        |

The RF model showcases high recall value, suggesting its ability to identify a significant portion of the failures effectively. However, its precision and F1 Score values are notably lower, indicating a higher rate of false positives. This suggests a potential for misclassification of instances, which could impact the model’s reliability in predictive maintenance tasks.

LightGBM showcases the highest precision and F1 Score values among all models. However, it exhibits the lowest recall value, indicating difficulty in correctly identifying true positives. In scenarios where accurately predicting failures is crucial, this model may prove inefficient due to its tendency to miss relevant instances.

CatBoost and GBM exhibit competitive recall rates, signifying their effectiveness in identifying relevant instances. However, their precision is relatively lower compared to LightGBM. This suggests challenges in distinguishing between positive and negative instances, potentially leading to inefficiencies in the predictions.

XGB showcases the best recall rate, indicating its effectiveness in identifying relevant instances in predictive maintenance tasks. However, it falls short in terms of precision and F1 Score values compared to some models. The resulting confusion matrix of the XGB model is depicted in Figure 4.1.

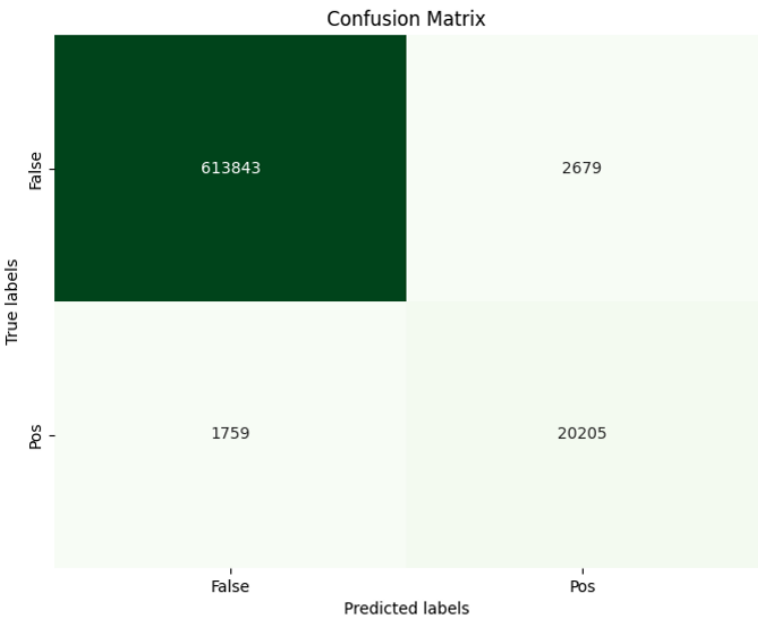


Figure 4.1: Confusion matrix of XGBoost model.

The Voting Model combined the strengths of multiple algorithms, aiming to enhance overall performance. Despite achieving impressive performance with all metrics above 90%, it lags behind individual models such as CatBoost and XGB in terms of recall. Nevertheless, it remains a robust and versatile model, demonstrating strong performance across various evaluation criteria.

The evaluation of these models revealed mixed performance across key metrics. RF lagged behind in terms of precision and F1 Score compared to the other models, indicating less satisfactory results. LightGBM performed exceptionally well in terms of precision and F1 Score but showed limitations in the recall metric, suggesting overall good performance with some trade-offs. On the other hand, models like XGB demonstrated excellent recall but lower precision and F1 Score, indicating effective identification of relevant instances despite some room for improvement in precision.

## 4.2 Rule-Based Adjustments

Applying rule-based adjustments significantly enhanced the data-driven model's performance. By integrating domain-specific rules, the system improved its handling of rare but critical failure scenarios. For instance, rules were implemented to address conditions where certain sensor readings, when combined, indicated an impending failure that the model alone did not detect.

Table 4.2 presents the considered performance metrics before and after applying the rule-based adjustments. The improvements in recall, precision, and F1 Score across different models highlight the added value of expert knowledge in enhancing predictive accuracy.

Table 4.2: Comparison of the implemented Machine Learning models performance before and after incorporating knowledge-based rules.

|                         | Recall (%)    | F1 Score (%)  | Precision (%) | Accuracy (%)  |
|-------------------------|---------------|---------------|---------------|---------------|
| RF - before rules       | 90.685        | 64.655        | 50.236        | 96.586        |
| RF - after rules        | 90.999        | <b>93.093</b> | 95.285        | 99.535        |
| XGB - before rules      | 91.991        | 90.104        | 88.293        | 99.305        |
| XGB - after rules       | 91.991        | 90.471        | 89.001        | 99.333        |
| LGBM - before rules     | 88.786        | 92.144        | 95.767        | 99.479        |
| LGBM - after rules      | 90.748        | 92.232        | <b>95.855</b> | <b>99.547</b> |
| CatBoost - before rules | 91.413        | 89.303        | 87.288        | 99.247        |
| CatBoost - after rules  | 91.413        | 90.771        | 90.137        | <b>99.547</b> |
| GBM - before rules      | 91.003        | 89.576        | 88.193        | 98.886        |
| GBM - after rules       | <b>92.242</b> | 90.245        | 88.333        | 98.950        |
| Voting - before rules   | 91.003        | 91.055        | 91.107        | 99.059        |
| Voting - after rules    | 91.413        | 91.278        | 91.143        | 99.081        |

Incorporating domain knowledge through the application of rules has notably enhanced the predictive maintenance system, particularly in refining the model’s predictive capabilities. Implementing these rules provided a hybrid approach, utilising expert insights to refine the model’s predictions, ensuring a closer fit with the specific requirements of maintenance operations. The system’s performance, as evidenced by various evaluation metrics, demonstrated substantial improvement following the integration of these rules.

Among all evaluated models, the RF model demonstrated the most significant performance improvement after implementing rules, achieving an increase of 28 percentage points in the F1 score and an impressive 45 percentage points in the precision metric. These results highlight the success and effectiveness of integrating domain-specific rules in enhancing model performance.

The RF confusion matrix before and after the implementation of rules can be seen in Figure 4.2 and 4.3 respectively. A closer examination of the RF confusion matrix before and after the implementation of rules reveals the concrete improvements resulting from the integration of domain knowledge. In the pre-rule implementation confusion matrix, the RF exhibited numerous misclassifications, leading to discrepancies in the prediction outcomes. However, post-rule integration, these misclassifications have been notably reduced, indicating a more accurate alignment of model predictions with actual maintenance requirements.

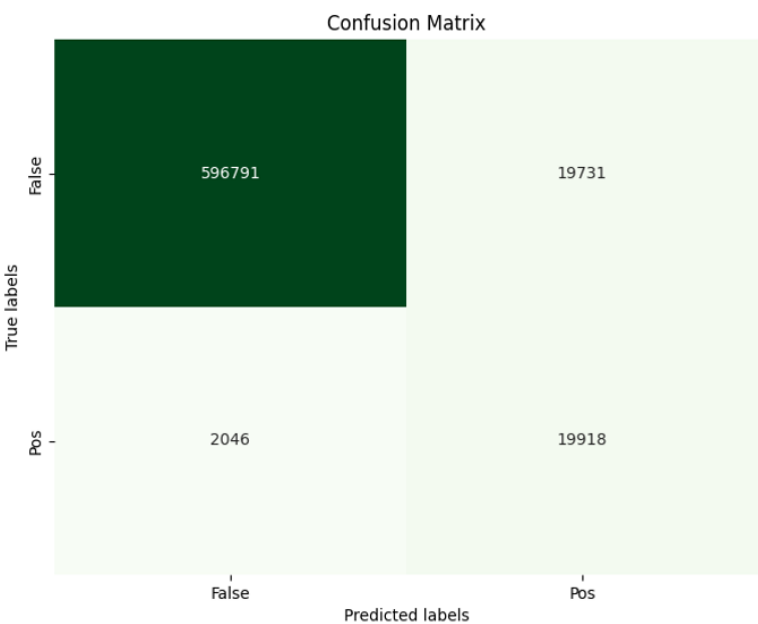


Figure 4.2: Confusion matrix of Random Forest model, before the implementation of rules.



Figure 4.3: Confusion matrix of Random Forest model, after the implementation of rules.

This marked enhancement in the performance of the RF model highlights the benefit of incorporating domain knowledge into predictive maintenance systems. By exploiting expert insights through the formulation of rules, the system can effectively augment its predictive capabilities, thereby facilitating more informed decision-making and proactive maintenance strategies.

In the RF system, Rule 1 was not applied, while Rule 2 was applied 18,832 times, resulting in 18,742 correct predictions. Rule 3 was applied and resulted in 159 correct predictions. Thus, Rule 2 achieved a 99.5% accuracy rate, while Rule 3 achieved a perfect 100% accuracy rate.

Overall, the integration of these rules, as illustrated in Table 4.3, contributed to enhanced performance. While Rule 1 remained unapplied in any model, its potential significance in addressing edge cases can be very important. Rule 2 emerged as the most frequently applied, achieving an impressive accuracy rate of approximately 99%. Meanwhile, Rule 3 consistently delivered perfect accuracy, demonstrating its robustness. Collectively, these rules significantly enhanced the models' predictive capabilities.

Table 4.3: Evaluation metrics of rule application in Machine Learning models.

| Rule | Applied | Correct | Accuracy |
|------|---------|---------|----------|
| 1    | 0       | 0       | -        |
| 2    | 19,831  | 19,651  | 99%      |
| 3    | 1,042   | 1,042   | 100%     |

When selecting the model for implementation into the final system, the hybrid approach, which incorporated domain knowledge through rules alongside the GBM model, stood out for its exceptional recall performance. This ability to accurately identify relevant instances enhances overall system reliability, making it a strategic choice for maximising predictive accuracy in maintenance tasks. The incorporation of domain knowledge notably enhances the recall metric by 1.2 percentage points, further solidifying its effectiveness.

Every model experienced benefits from integrating domain knowledge through rules, resulting in notable improvements in their weakest performance metrics. For example, LightGBM, which exhibits the lowest recall score, increased its recall by 2 percentage points to achieve 90.748%. Similarly, CatBoost, which initially showed one of the lowest precision scores, improved by 3 percentage points, achieving 90.137% in the precision metric.

### 4.3 Model Explainability

LIME was integrated into the system through a textual approach, as illustrated in Figure 4.4.

| Feature                          | Score (?)             |
|----------------------------------|-----------------------|
| Reservoirs <= 8.63               | 0.3815799883640912    |
| HI <= 8.44                       | -0.2536118421869289   |
| 56.08 < Oil Temperature <= 61.27 | -0.056083398380746705 |
| -0.02 < Dv Pressure <= -0.02     | 0.009759769967159652  |
| -0.01 < Tp2 <= -0.01             | 0.005541741663219665  |
| Motor Current <= 0.04            | 0.0014123738661009145 |
| Towers <= 1.00                   | 0.0                   |

Figure 4.4: An example of a LIME explanation for a failure prediction.

In LIME, the larger the "contribution score", the more significant the feature is for the prediction. A positive score indicates that the feature contributes to a prediction of failure, while a negative score suggests that it contributes to a prediction of no failure. By displaying these explanations, the system's transparency is enhanced, allowing users to validate its accuracy, thereby building trust in the model's reliability and performance.

The rules provided by LIME may not always align exactly with those identified during the conducted data analysis aimed at identifying rules for implementation in the system. This discrepancy arises because LIME evaluates each sample individually as a local method, whereas the data analysis examines the entire dataset collectively.

Moreover, LIME explanations empower domain experts to not only create new rules but also refine existing ones based on insights derived from the model's results. By understanding the most influential features in predictions, users can fine-tune rules to enhance predictive accuracy and relevance.

In critical domains like predictive maintenance, where operational decisions are critical, LIME facilitates informed decision-making by providing clear, interpretable insights from ML models. This capability minimises uncertainty and boosts confidence in utilising ML outputs effectively.

## 4.4 Graphical User Interface

The UI serves as a central hub for users to interact with the developed system, offering a range of functionalities to streamline operational processes. Users can effortlessly submit failure reports, view predictions, and access detailed explanations, all in real-time. This real-time access ensures that users are always informed of the system's insights and can make informed decisions promptly.

One of the notable features of the developed UI is its responsiveness, ensuring rapid data retrieval and display. This responsiveness is particularly crucial in time-sensitive scenarios, where quick access to information can significantly impact decision-making and operational outcomes.

Moreover, the UI is integrated with the database, ensuring robust data storage and retrieval mechanisms. This integration enables the interface to access and present real-time data accurately, providing users with up-to-date insights into system performance and maintenance requirements.

In Figures 4.5, 4.6, 4.7, 4.8, 4.9, 4.10 and 4.11 the developed UI is illustrated.

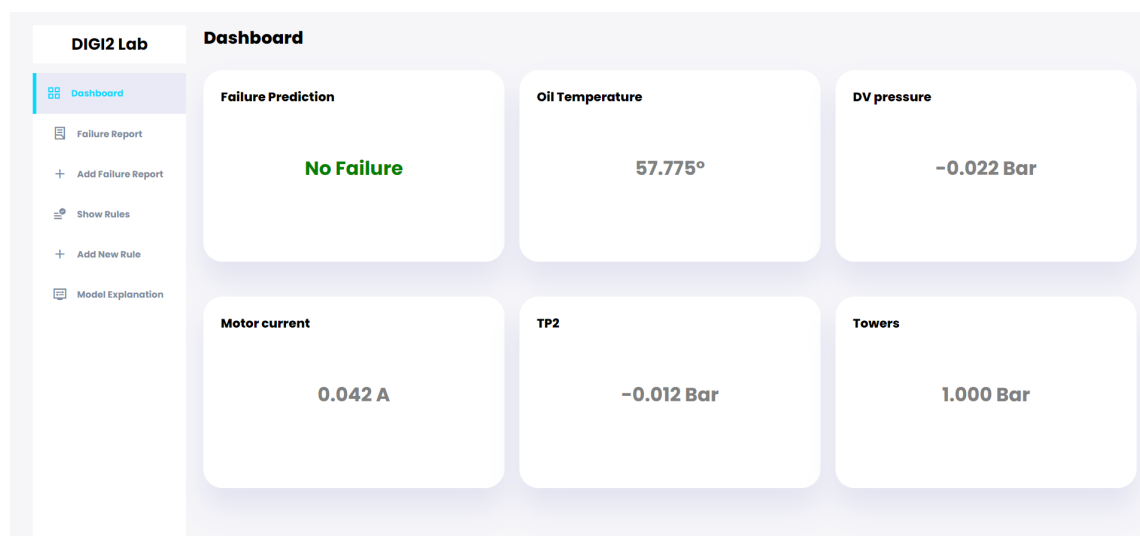
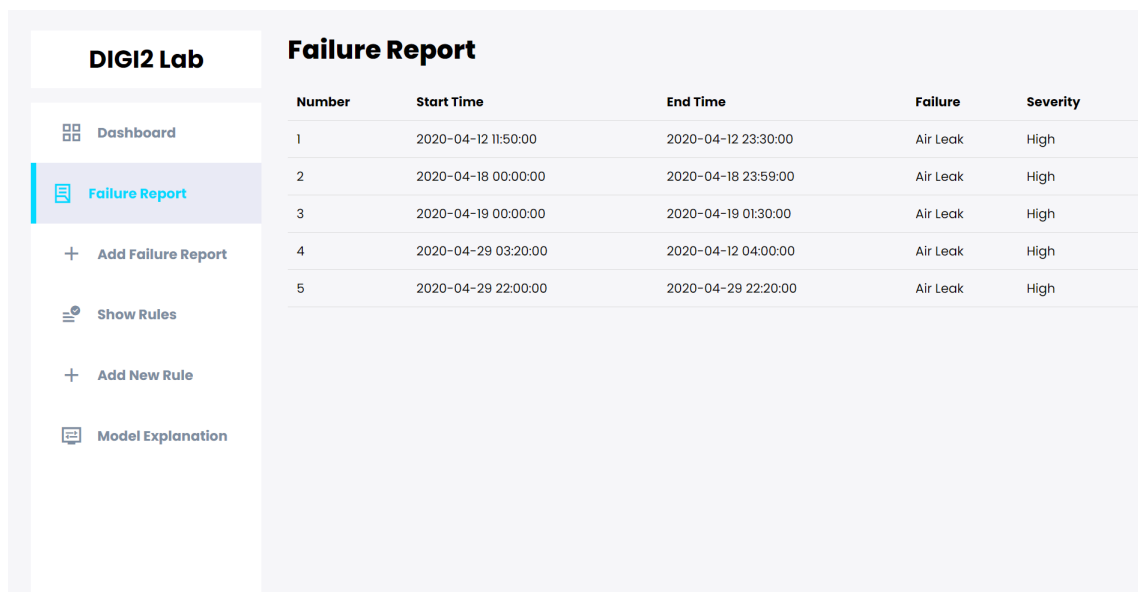
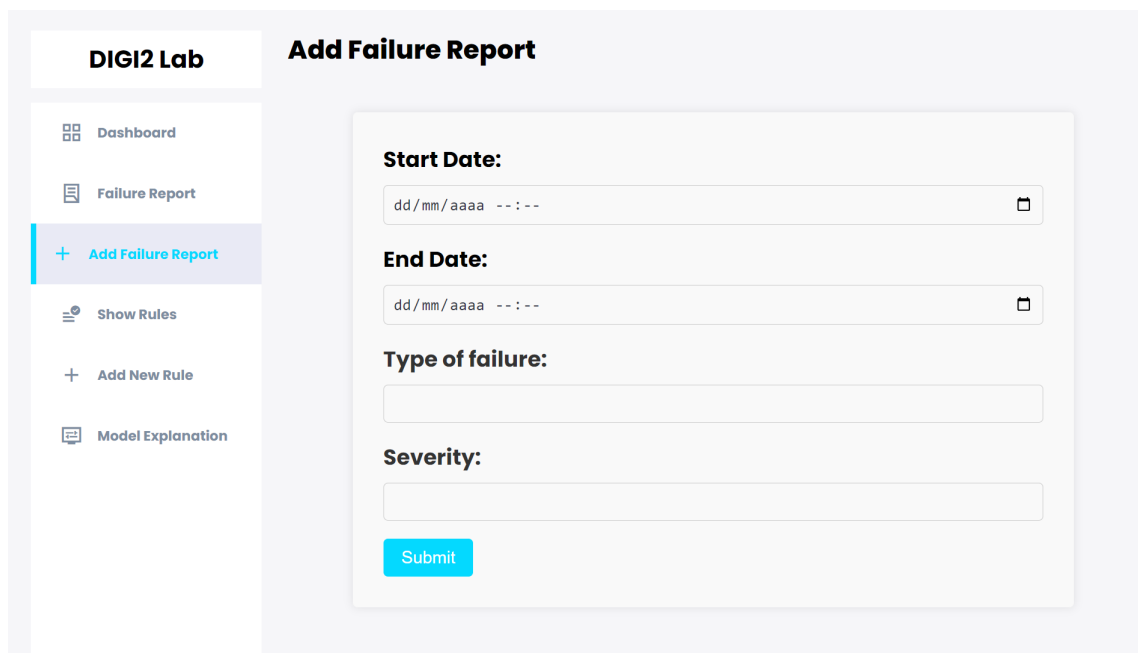


Figure 4.5: "Dashboard" page, in the web application.



| Number | Start Time          | End Time            | Failure  | Severity |
|--------|---------------------|---------------------|----------|----------|
| 1      | 2020-04-12 11:50:00 | 2020-04-12 23:30:00 | Air Leak | High     |
| 2      | 2020-04-18 00:00:00 | 2020-04-18 23:59:00 | Air Leak | High     |
| 3      | 2020-04-19 00:00:00 | 2020-04-19 01:30:00 | Air Leak | High     |
| 4      | 2020-04-29 03:20:00 | 2020-04-12 04:00:00 | Air Leak | High     |
| 5      | 2020-04-29 22:00:00 | 2020-04-29 22:20:00 | Air Leak | High     |

Figure 4.6: "Failures Reports" page, in the web application.



**Start Date:**

dd/mm/yyyy --:--

**End Date:**

dd/mm/yyyy --:--

**Type of failure:**

**Severity:**

Submit

Figure 4.7: "Add Failure Reports" page, in the web application.

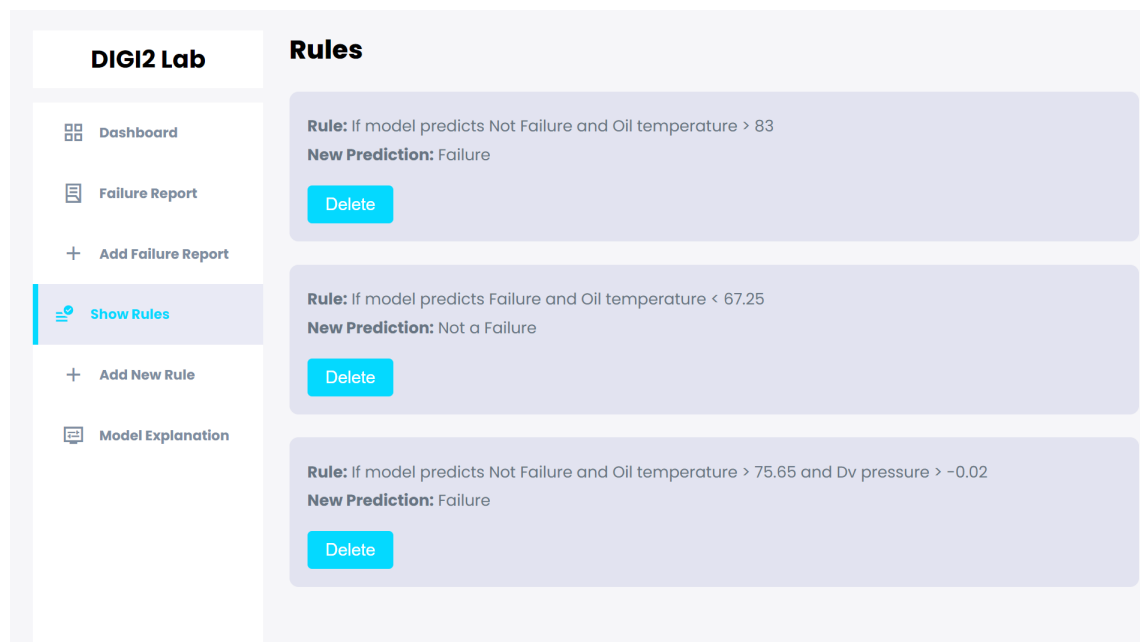


Figure 4.8: "Rules" page, in the web application.

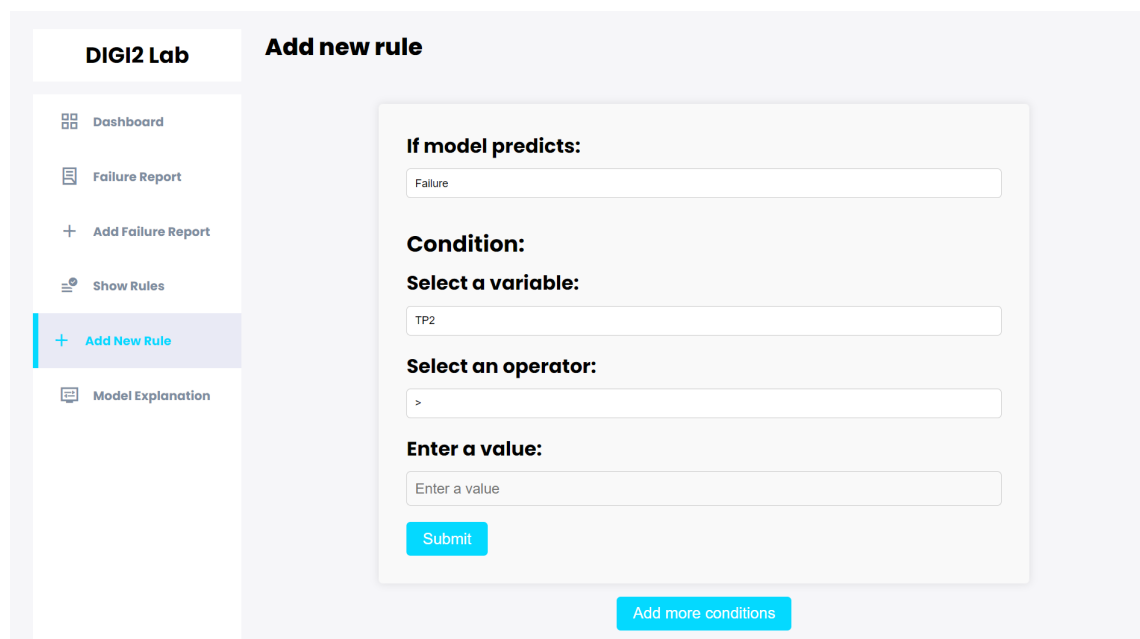


Figure 4.9: "Add New Rules" page, in the web application.

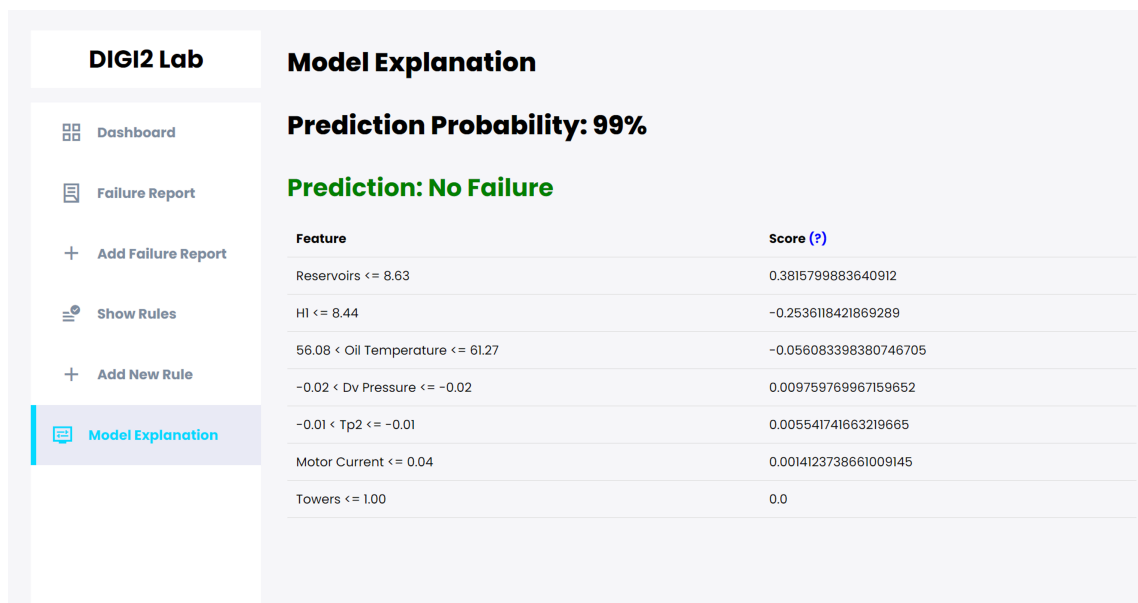


Figure 4.10: "Model Explanation" page, in the web application.

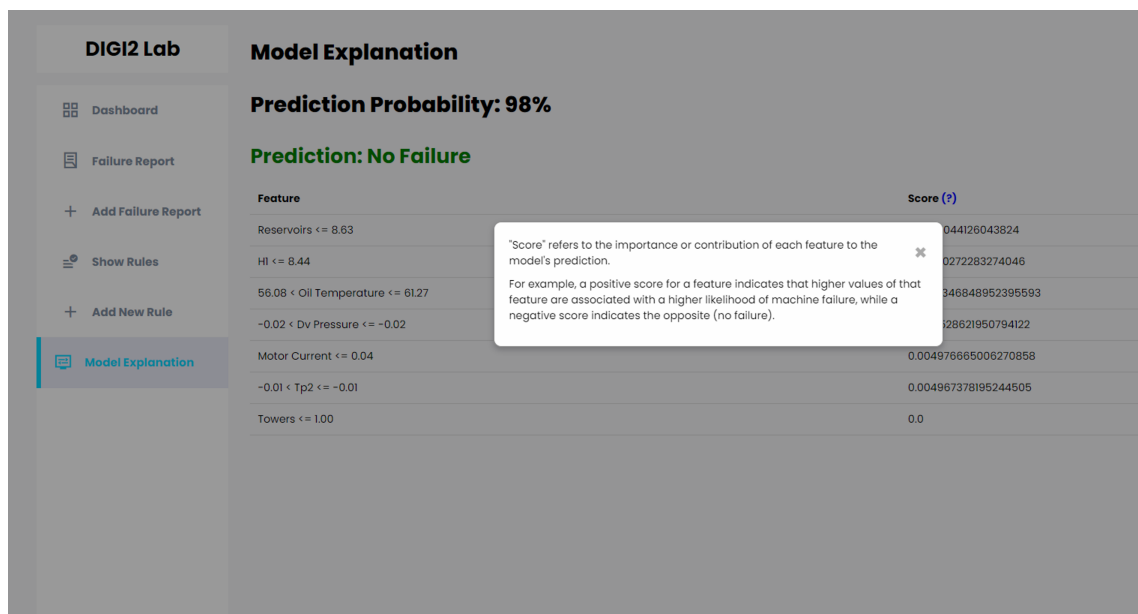


Figure 4.11: "Model Explanation" page, with open pop-up.

## 4.5 Prediction and Explanation Time

To evaluate the developed application, a script was created to simulate a publisher that sends MQTT messages every ten seconds. The dataset is contained within an Excel file, and each row is sent at ten-second intervals to match the timestamps in the dataset. This approach ensures that the application can handle periodic data updates in real time, effectively simulating a realistic data publishing environment. The application uses an MQTT subscriber to subscribe to the topic

to receive the dataset. When an MQTT message is received, the application starts executing its designated processes.

The duration values were calculated by averaging each metric over 100 iterations, utilising the time function from the Python library.

The system, employing the GBM model, demonstrated an average prediction generation time of 24.8 milliseconds per instance. This encompasses receiving an MQTT message with data, performing necessary pre-processing steps such as feature engineering, generating a prediction using the pre-processed data, and applying the rules. This rapid prediction time ensures the system can provide timely alerts, allowing for rapid intervention and mitigation measures. Such efficiency is critical in predictive maintenance tasks, where immediate action can prevent costly downtime and equipment failures, thereby significantly enhancing operational reliability.

Generating prediction explanations using LIME averaged 233.8 milliseconds per instance. The average time to send data to the database was 475.1 milliseconds. This quick data transmission ensures that the latest information is readily available for analysis and decision-making, enhancing the overall responsiveness and effectiveness of the predictive maintenance system.

The efficiency of different system components can be observed in Table 4.4:

Table 4.4: Time taken for various system tasks.

| System Task            | Time Taken (milliseconds) |
|------------------------|---------------------------|
| Prediction Generation  | 24.8                      |
| Explanation Generation | 233.8                     |
| Database Update        | 475.1                     |

In summary, the combination of rapid prediction times, explanation generation durations, and efficient data transmission contributes to a robust and reliable predictive maintenance system. Users can promptly act on predictions and understand the underlying reasons without significant delays, which is crucial in time-sensitive environments. These findings underscore the efficiency and effectiveness of the predictive maintenance system in handling prediction and explanation tasks.

4.6 System Validation

As part of the validation process for the hybrid predictive failure model developed in this project, three experts in the field were consulted, including a representative from "Metro do Porto". Their combined insights highlight the practical applications, benefits, and potential areas for improvement in the model. The experts validated the model through a video demonstration that explained its functionality and provided feedback.

The experts emphasised the importance of a hybrid predictive model in failure identification due to its capability to integrate specialised knowledge from operators with data-driven techniques. This integration enhances the system's responsiveness and accuracy in identifying potential failures, which is crucial for industrial efficiency and operational continuity.

The experts found this hybrid approach very interesting. They noted that the intuitive dashboard design provides a comprehensive overview of the states of components, making it easy for users to monitor and manage. Additionally, they highlighted that the simplicity of inputting new failures and rules helps lower barriers to industrial adoption. The automatic model retraining feature upon submitting a failure report was particularly well-received, as it enhances the system's efficiency and responsiveness. The drag-and-drop functionality for prioritisation was also praised for its user-friendliness. Furthermore, the experts emphasised that through the user interface, rules can be prioritised and modified, offering flexibility to adapt the model to various scenarios with different operating conditions.

In summary, the expert feedback underscores the substantial benefits of integrating human expertise with data-driven methods in a hybrid predictive model. The system's flexibility, user-friendly interface, and automatic retraining capabilities make it a practical and valuable tool for improving predictive maintenance in industrial settings. Below is a summary of the individual feedback from each expert:

- **Expert from Metro do Porto:** Highlighted the interesting nature of the approach and praised the prioritisation and implementation of rules.
- **Expert from STAR Institute:** Emphasized the intuitive dashboard design and its effectiveness in monitoring components. Appreciated the ease of inputting new failures and rules, and the user-friendliness of the drag-and-drop functionality.
- **Expert from Harms & Wende:** Valued the combination of expert knowledge with data-driven methodologies and the automatic model retraining. Highlighted the low barriers for industrial adoption due to the simple input process for new failures and rules.

## 4.7 Summary

Chapter 4 detailed the performance and results of the failure prediction and reporting system. Six models, including RF, XGB, LightGBM, GBM, CatBoost, and an ensemble model, were evaluated using metrics such as recall, precision, F1 Score, and accuracy. LightGBM achieved the highest precision and F1 Score, while XGB excelled in the recall metric. The ensemble model demonstrated strong overall performance, providing a comprehensive comparison that highlights the strengths and weaknesses of each model.

The integration of domain-specific rules substantially boosted model performance by as much as 28 percentage points in the F1 Score and 45 percentage points in the precision metric. Applying these rules improved recall, precision, and F1 Scores across different models, highlighting the

value of expert knowledge with ML. Among these approaches, the GBM model combined with rules stood out for achieving the highest recall values.

Employing LIME enhanced the model's transparency, enabling users to validate and refine its predictions with greater confidence. The developed UI facilitates easy interaction with the system, enabling users to submit reports, view predictions, and access explanations in real-time. The interface's responsiveness and integration with the database ensure that users receive up-to-date information, aiding in effective decision-making.

The system demonstrated efficient performance, with an average prediction generation time of 24.8 milliseconds and an explanation generation time of 233.8 milliseconds. The rapid processing times ensure timely alerts and actionable insights for predictive maintenance. Validation with industrial personnel confirmed the system's practical utility and usability. The intuitive UI, ease of inputting new data, automatic model retraining, and user-friendly prioritisation features were particularly well-received.

In conclusion, integrating domain-specific rules with ML models significantly enhances predictive maintenance capabilities, providing a robust, transparent, and efficient system that meets industrial needs.



## Chapter 5

# Conclusion and Future Work

The primary objective of this dissertation was to develop a hybrid failure prediction to enhance operational efficiency and reliability in industrial environments. Through rigorous evaluation and comparison, the GBM algorithm, augmented with domain knowledge through rule integration, emerged as the most suitable choice for predictive modelling. This approach demonstrated superior performance, particularly in terms of recall, which is critical for accurately identifying potential failures and minimising false negatives in operational settings.

The results indicate that the developed system successfully achieves its goal by integrating domain knowledge with data-driven methods, significantly enhancing predictive accuracy. This hybrid approach has proven effective in accurately predicting failures and promptly reporting them, minimising downtime and improving overall operational performance. The integration of Explainable Artificial Intelligence techniques enhanced the system's transparency, providing users with clear insights into the decision-making process behind each prediction.

The system features an interface that allows users to integrate new rules, adjust their priorities, and delete existing ones. This flexibility ensures that the system can be easily updated to reflect the latest domain knowledge and operational priorities. Additionally, users can submit failure reports through the interface, prompting the system to retrain the model with newly available data. This continuous learning process allows the model to adapt to new patterns and improve its predictive capabilities.

Validation by industrial personnel ensured that the system met practical needs and usability standards. They found that the intuitive dashboard design and hybrid approach facilitate easy monitoring and management. Additionally, the simplified input for new failures and rules lowers adoption barriers, while the automatic model retraining feature enhances both efficiency and user-friendliness.

This work contributes to the advancing knowledge of hybrid approaches in predictive maintenance systems. A scientific paper detailing the findings and methodologies has been submitted to the ICINCO conference. Details can be found in Appendix A. Additionally, a GitHub repository has been created where the produced code and a video demonstration of the web interface can be accessed at <https://github.com/miguelalmeida8/HyPredictor-Framework>.

All the established research questions were addressed, providing valuable insights into the integration of expert knowledge into predictive maintenance models. To answer the first research question, expert knowledge was effectively integrated into data-driven predictive maintenance models by using rule-based systems. For the second, integrating expert knowledge contributed to sustainability by predicting potential future failures, thus minimising waste and resource consumption. Combining data-driven methods with knowledge-based rules, as addressed in the third research question, improved the overall efficiency and effectiveness of maintenance strategies. For the final research question, while challenges such as utilising the knowledge effectively were identified, significant opportunities for enhanced model interpretability and improved decision-making were also noted. Addressing these challenges will lead to more adaptive maintenance systems, exploiting both data-driven insights and domain-specific knowledge.

Future work should focus on ensuring the scalability of the failure prediction and reporting system across various industrial environments. Additionally, developing the capability to import failure reports from log files would be a valuable functionality to explore.

In conclusion, this project has established a robust foundation for a reliable failure prediction and reporting system and demonstrated the substantial advantages of integrating hybrid approaches that combine data-driven with knowledge-based methods. This hybrid approach holds significant promise for enhancing operational efficiency and reliability within industrial environments, paving the way for broader implementation and adoption.

# References

- [1] United Nations. THE 17 GOALS | Sustainable Development — sdgs.un.org. <https://sdgs.un.org/goals>, 2024. [Accessed 15-06-2024].
- [2] Yannick Wilhelm, Peter Reimann, Wolfgang Gauchel, and Bernhard Mitschang. Overview on hybrid approaches to fault detection and diagnosis: Combining data-driven, physics-based and knowledge-based models. volume 99, pages 278–283. Elsevier B.V., 2021.
- [3] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.
- [4] Metro Porto. Porto Metro - Maps, timetables and updated prices — porto.travel. <https://porto.travel/metro/>, 2022. [Accessed 12-06-2024].
- [5] Joerg Leukel, Julian González, and Martin Riekert. Adoption of machine learning technology for failure prediction in industrial maintenance: A systematic review. *Journal of Manufacturing Systems*, 61:87–96, 2021.
- [6] Narjes Davari, Bruno Veloso, Rita P Ribeiro, Pedro Mota Pereira, and João Gama. Predictive maintenance based on anomaly detection using deep learning for air production unit in the railway industry. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10. IEEE, 2021.
- [7] Tiago Zonta, Cristiano André da Costa, Rodrigo da Rosa Righi, Miromar José de Lima, Eduardo Silveira da Trindade, and Guann Pyng Li. Predictive maintenance in the industry 4.0: A systematic literature review. *Computers and Industrial Engineering*, 150, 12 2020.
- [8] Hongfei Li, Dhaivat Parikh, Qing He, Buyue Qian, Zhiguo Li, Dongping Fang, and Arun Hampapur. Improving rail network velocity: A machine learning approach to predictive maintenance. *Transportation Research Part C: Emerging Technologies*, 45:17–26, 2014.
- [9] Mikel Canizo, Enrique Onieva, Angel Conde, Santiago Charramendieta, and Salvador Trujillo. Real-time predictive maintenance for wind turbines using big data frameworks. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 70–77, 2017.
- [10] Kevin Leahy, Colm Gallagher, Peter O’Donovan, Ken Bruton, and Dominic T.J. O’Sullivan. A robust prescriptive framework and performance metric for diagnosing and predicting wind turbine faults based on scada and alarms data with case study. *Energies*, 11, 2018.

- [11] Gian Antonio Susto, Andrea Schirru, Simone Pampuri, Seán McLoone, and Alessandro Beghi. Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, 11:812–820, 6 2015.
- [12] R. Lily Hu, Kevin Leahy, Ioannis C. Konstantakopoulos, David M. Auslander, Costas J. Spanos, and Alice M. Agogino. Using domain knowledge features for wind turbine diagnostics. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 300–307, 2016.
- [13] Francesca Calabrese, Alberto Regattieri, Marco Bortolini, Mauro Gamberi, and Francesco Pilati. Predictive maintenance: a novel framework for a data-driven, semi-supervised, and partially online prognostic health management application in industries. *Applied Sciences (Switzerland)*, 11, 4 2021.
- [14] Angelos Angelopoulos, Emmanouel T. Michailidis, Nikolaos Nomikos, Panagiotis Trakadas, Antonis Hatziefremidis, Stamatis Voliotis, and Theodore Zahariadis. Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects. *Sensors*, 20(1), 2020.
- [15] Zeki Murat Çınar, Abubakar Abdussalam Nuhu, Qasim Zeeshan, Orhan Korhan, Mohammed Asmael, and Babak Safaei. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability*, 12(19), 2020.
- [16] Andrew Kusiak and Anoop Verma. A data-mining approach to monitoring wind turbines. *IEEE Transactions on Sustainable Energy*, 3:150–157, 1 2012.
- [17] Kunru Chen, Sepideh Pashami, Yuantao Fan, and Slawomir Nowaczyk. Predicting air compressor failures using long short term memory networks. In *Progress in Artificial Intelligence*, pages 596–609, Cham, 2019. Springer International Publishing.
- [18] Pier Francesco Orrù, Andrea Zoccheddu, Lorenzo Sassu, Carmine Mattia, Riccardo Cozza, and Simone Arena. Machine learning approach using mlp and svm algorithms for the fault prediction of a centrifugal pump in the oil and gas industry. *Sustainability (Switzerland)*, 12, 6 2020.
- [19] Filipe Alves, Hasmik Badikyan, H.J. António Moreira, João Azevedo, Pedro Miguel Moreira, Luís Romero, and Paulo Leitão. Deployment of a smart and predictive maintenance system in an industrial case study. In *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*, pages 493–498, 2020.
- [20] Shili Xiang, Dong Huang, and Xiaoli Li. A generalized predictive framework for data driven prognostics and diagnostics using machine logs. In *TENCON 2018 - 2018 IEEE Region 10 Conference*, pages 0695–0700, 2018.
- [21] Adrian Binding, Nicholas Dykeman, and Severin Pang. Machine learning predictive maintenance on data in the wild. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pages 507–512, 2019.
- [22] Qingqiang Sun and Zhiqiang Ge. A survey on deep learning for data-driven soft sensors. *IEEE Transactions on Industrial Informatics*, 17:5853–5866, 9 2021.
- [23] Yuanfang Chi, Yanjie Dong, Z. Jane Wang, F. Richard Yu, and Victor C.M. Leung. Knowledge-based fault diagnosis in industrial internet of things: A survey. *IEEE Internet of Things Journal*, 9:12886–12900, 8 2022.

- [24] Antonin Gay, Benoit Iung, Alexandre Voisin, Phuc Do, Remi Bonidal, and Ahmed Khelassi. A short review on the integration of expert knowledge in prognostics for phm in industrial applications. pages 286–292. Institute of Electrical and Electronics Engineers Inc., 2021.
- [25] Alberto Diez-Olivan, Jose A. Pagan, Ricardo Sanz, and Basilio Sierra. Data-driven prognostics using a combination of constrained k-means clustering, fuzzy modeling and lof-based score. *Neurocomputing*, 241:97–107, 2017.
- [26] Arka Daw, Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. 10 2017.
- [27] B. Satish and N.D.R. Sarma. A fuzzy bp approach for diagnosis and prognosis of bearing faults in induction motors. In *IEEE Power Engineering Society General Meeting, 2005*, pages 2291–2294 Vol. 3, 2005.
- [28] A.K. Garga, K.T. McClintic, R.L. Campbell, Chih-Chung Yang, M.S. Lebold, T.A. Hay, and C.S. Byington. Hybrid reasoning for prognostic learning in cbm systems. In *2001 IEEE Aerospace Conference Proceedings (Cat. No.01TH8542)*, volume 6, pages 2957–2969 vol.6, 2001.
- [29] Yanrui Li and Chunjie Yang. Domain knowledge based explainable feature construction method and its application in ironmaking process. *Engineering Applications of Artificial Intelligence*, 100:104197, 2021.
- [30] Dang Minh, H Xiang Wang, Y Fen Li, and Tan N Nguyen. Explainable artificial intelligence: a comprehensive review. *Artificial Intelligence Review*, pages 1–66, 2022.
- [31] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1), 2021.
- [32] Oscar Serradilla, Ekhi Zugasti, Carlos Cernuda, Andoitz Aranburu, Julian Ramirez de Okariz, and Urko Zurutuza. Interpreting remaining useful life estimations combining explainable artificial intelligence and domain knowledge in industrial machinery. In *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8, 2020.
- [33] SG Zeldam. Automated failure diagnosis in aviation maintenance using explainable artificial intelligence (xai). Master’s thesis, University of Twente, 2018.
- [34] Veloso Bruno Ribeiro Rita Davari, Narjes and Joao Gama. MetroPT-3 Dataset. UCI Machine Learning Repository, 2023. DOI: <https://doi.org/10.24432/C5VW3R>.



## Appendix A

# Conference Paper

The scientific paper titled "HyPredictor - Hybrid Failure Prognosis Approach Combining Data-Driven and Knowledge-Based Methods" was submitted to the ICINCO 2024 conference.

The paper abstract is provided below:

"In modern manufacturing, marked by an unprecedented surge in data generation, utilising this wealth of information to enhance company performance has become essential. Within the industrial landscape, one of the significant challenges is equipment failures, which can result in substantial financial losses and wasted time and resources. This work presents the HyPredictor framework, a comprehensive failure prediction and reporting system designed to enhance the reliability and efficiency of industrial operations by leveraging advanced machine learning techniques and domain knowledge. Six machine learning algorithms were evaluated for failure prediction. The predictions from the algorithms are then refined using rule-based adjustments derived from domain knowledge. Additionally, Explainable AI (XAI) techniques were incorporated, as well as the capability of users to customise the system with their own rules and submit failure reports, prompting model retraining and continuous improvement. Integrating domain-specific rules improved the performance by up to 28 percentage points in the F1 Score metric in some prediction models, with the best hybrid approach achieving an F1 Score of 90% and a Recall of 92% in failure prediction. This adaptive, hybrid approach improves prediction accuracy and fosters proactive maintenance, significantly reducing downtime and operational costs."