UNIVERSIDADE PORTUCALENSE INFANTE D. HENRIQUE

Explainable Predictive Maintenance: Autoencoders and Rule-based Explainer

Ricardo Queiroz

Mestrado Ciência dos Dados

Supervisor: Professor Doutor Bruno Veloso

Co-Supervisor: Professor Doutor João Gama

September 28, 2023

Explainable Predictive Maintenance: Autoencoders and Rule-based Explainer

Ricardo Queiroz

Mestrado Ciência dos Dados

September 28, 2023

Abstract

In recent years, the increase in artificial intelligence's prevalence and sophistication has accelerated its use in various fields. However, this growth has come with challenges, especially in the form of complex, opaque models. These models, frequently called "black boxes," have remarkable predictive power but tend to obscure the logic behind their decisions, making them less trustworthy and adaptable in critical contexts. This dissertation explores the creation of a black box LSTM Autoencoder for predictive maintenance, utilizing the potential of explainable artificial intelligence (XAI). In particular, we employ and compare SHAP (SHapley Additive exPlanations) and AMRules, two XAI techniques, to explicate the outcomes and unravel the inner workings of this complex model. In this work, we determine the anomaly detection performance using three datasets: the first and second versions of the MetroPT and the Nasa datasets. We further examine the outcomes of each XAI method employed, highlighting their respective advantages and limitations. In our investigation, we concluded that it is possible to explain the predictions generated by a black box model without compromising its performance.

Keywords: LSTM-Autoencoder, SHAP, AMRules, Predictive Maintenance, XAI, Anomaly Detection

Resumo

Nos últimos anos, o aumento da prevalência e sofisticação da inteligência artificial tem acelerado o seu uso numa grande variedade de campos. No entanto, este crescimento não veio sem desafios, especialmente na forma de modelos complexos e opacos. Estes modelos, que são frequentemente referidos como "caixas negras", têm um poder preditivo notável, mas tendem a obscurecer a lógica por trás de suas decisões, tornando-os menos confiáveis e adaptáveis em contextos críticos. Esta tese explora a criação de um LSTM Autoencoder para manutenção preditiva, utilizando o potencial de inteligência artificial explicável (XAI). Em particular, empregamos e comparamos SHAP (SHapley Additive exPlanations) e AMRules, duas técnicas XAI, para explicar os resultados e desvendar as operações internas deste modelo complexo. Neste trabalho, determinamos o desempenho da detecção de anomalias usando três conjuntos de dados, as primeira e segunda versão do dataset MetroPT, bem como o dataset da NASA. Examinamos ainda os resultados de cada método XAI empregado, destacando suas respectivas vantagens e limitações. Na nossa investigação, concluímos que é possível fornecer explicações para as previsões geradas por um modelo de caixa negra sem comprometer o seu desempenho.

Keywords: LSTM-Autoencoder, SHAP, AMRules, Predictive Maintenance, XAI, Anomaly Detection

Acknowledgements

Ao Professor Doutor Bruno Veloso, que apoiou e orientou com dedicação, e ao Professor Doutor João Gama que co-orientou o desenvolvimento desta dissertação.

À minha família que sempre me apoiou, e aos meus colegas e amigos com quem partilhei estes últimos anos, Um grande obrigado.

Ricardo Queiroz

"The only true wisdom is in knowing you know nothing."

Socrates

Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial Intelligence
AMRules	Adaptive Model Rules
ANN	Artificial Neural Network
ARIMA	AutoRegressive Integrated Moving Average
AUPRC	Area Under the Precision-Recall Curve
AUROC	Area Under the Receiver Operating Characteristic
AURPC	Area Under the Receiver Operating Characteristic Curve
CC	Convolutional Layers
CNN	Convolutional Neural Network
EU	European Union
FPR	False Positive Rate
GDPR	General Data Protection Regulation
K-NN	K-Nearest Neighbors
LGBM	Light Gradient-Boosting Machine
LIME	Local Interpretable Model-Agnostic Explanations
LR	Linear Regression
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MDMC	Mean Degree of Metrics Change
ML	Machine Learning
MLE	Maximum Likelihood Estimation
MSE	Mean Squared Error
NASA	National Aeronautics and Space Administration
PDM	Predictive Maintenance
PVM	Preventive maintenance
R2F	Run-To-Failure
RF	Radom Forest
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SHAP	SHapley Additive Explanations
SVM	Support-Vector Machine
SWRL	Semantic Web Rule Language
TPR	True Positive Rate
XAI	Explainable artificial intelligence

Chapter 1

Introduction

Artificial intelligence (AI) has seen a notable rise in its use in recent years due to its shown efficacy in the real world. Consequently, AI has progressively gained importance in our daily routines. These technologies are being rapidly employed across various sectors, including education, finance, and healthcare. There is a big focus on developing more effective and accurate models in AI, which has increased complexity. The models become extremely difficult to comprehend and lose the ability to explain how outcomes are obtained. This type of model is known as the "Black Box Model."

The performance of these models can reach incredibly high levels of precision. However, this does not imply that the models are flawless and error-free. Due to the increased use of these algorithms in daily operations, their errors can have significant repercussions. For instance, it is possible to extrapolate that an error in an autonomous driving system could result in an accident that puts multiple lives at risk. Explaining the actions and the cause of the error is essential in these situations. The European Union has recognized the significant impact of artificial intelligence (AI) on our daily lives. As a result, it has implemented a policy inside the General Data Protection Regulation (GDPR) that grants citizens the right to request an explanation for algorithmic decisions. Additionally, the EU has emphasized the urgent necessity for human interpretability in the design of these algorithms. Another important aspect of artificial intelligence is that models' performance and behaviour depend significantly on the training data. If the data contains any bias, the predictions will reflect this, which may result in discrimination against particular groups. Thus, it is crucial that the predicted outcomes can be explained to detect biases.

For these reasons, the importance of Explainable artificial intelligence is increasing. This field aims to unravel the internal workings of black box models and provide insights into the decision-making process. This capability allows humans to comprehend the factors contributing to model predictions, increasing their trust in the results.

1.1 Context

Our research will focus on explaining the black box models within the context of predictive maintenance. This field has gained significant traction across several industries due to the pressing need to prevent equipment malfunctions, enhance maintenance planning, and save operational expenses. Organizations attempt to anticipate and mitigate possible equipment failures by utilizing historical data and employing advanced analytics techniques. Machine learning models, specifically black box models, have gained significant attention due to their capacity to provide precise forecasts, allowing proactive maintenance measures to be implemented.

Industries such as manufacturing, energy, transportation, and healthcare greatly depend on efficiently operating complex machinery. Unplanned downtime due to equipment failure results in financial losses and negatively impacts productivity and safety. Traditional maintenance methods, which involve regular inspections or pre-established timetables, often demonstrate inefficiency and high expenses. The rise of machine learning, influenced by increased computer power and the abundance of data, has facilitated the development of more advanced maintenance procedures.

1.2 Motivation

Exploring the field of Explainable Machine Learning intrigued me because of its novelty and the difficulties it presented. It was a compelling personal and professional development opportunity, introducing me to various obstacles requiring problem-solving skills, because even highly accurate models can contain errors, the potential consequences of such errors cannot be underestimated. Consider the railroad industry as an illustration. A failure to anticipate a critical event could result in accidents, negatively impact customer satisfaction, and, in extreme cases, endanger lives. In these circumstances, comprehending why a failure went undetected is of the utmost significance. Nevertheless, the complexity of black-box models makes it difficult to comprehend how they operate and to identify errors. This is where explainability becomes crucial, as it permits detecting and correcting these problems. Moreover, explainability is crucial for maintaining people's trust by explicating the reasons behind a model's error. Often, it is necessary to maintain the confidence of stakeholders, in particular, by explaining the cause of a failed prediction. A further advantage of a model that can be explained is its ability to identify the fundamental causes of failure. By comprehending predictions, we can gather invaluable insights that aid in preventing recurring malfunctions. Consensually this new field has a huge importance nowadays, and I am eager to contribute to it with this dissertation.

1.3 Objectives

The primary objective of this study is to establish the feasibility of incorporating explainability techniques into black box predictive maintenance models while preserving their accuracy. To achieve this, we will utilize the datasets from MetroPT and NASA. Additionally, we will employ

explainability techniques to gain a deeper understanding of the learned features and identify the key factors within the input data that have most impacted the decision.

1.4 Dissertation Structure

In the first chapter, we discuss the importance of the explainability of black box models and predictive maintenance. In the second chapter, we conduct a literature review in which we provide various models of machine learning applied to predictive maintenance and their corresponding outcomes, as well as different explainability models and their performance based on prior studies. In the third chapter, we present the datasets used in this study, describe our proposed predictive maintenance model, describe the evaluation methodology, and define the model's explainability methods. In the fourth chapter, we present the study's results regarding our model's performance in detecting anomalies and its explainability. In chapter five, we present the conclusions and recommendations for further research.

Chapter 2

Literature Review

In this chapter, we present the literature review, starting by describing what predictive maintenance is and its different categorizations, explaining the various machine learning models used in this field, and presenting the related research. Afterwards, we explain what explainable artificial intelligence is and its different types, present different XAI methods, and analyze prior research in this field. We conclude with a summary of this chapter.

2.1 Predictive Maintenance

Since the advent of large-scale manufacturing, maintaining the fundamental condition of assets has played a crucial role in sustaining productivity. In order to provide the industrial sector with production systems that are safe, environmentally friendly, and able to create high-quality components, maintenance is becoming even more vital as automation and industry competition rise [6].

According to the author Susto et al. [40], strategies for maintenance management may be categorized into three primary groups in order of increasing complexity and effectiveness:

- Run-to-failure (R2F) is only performed after the equipment fails. This is the simplest approach to maintenance, but it is also the least effective since it necessitates a production stoppage and the replacement of components, resulting in higher costs than those associated with preventative measures.

- Preventive maintenance (PvM) is a maintenance strategy executed periodically according to a plan based on time or process iterations to anticipate equipment breakdowns. It is an effective strategy for avoiding failures. However, more corrective steps are needed, resulting in increased operational expenses and the efficient use of resources.

- Predictive Maintenance (PdM) employs predictive techniques to anticipate when maintenance operations are required. Monitoring a machine's or process's integrity continuously enables maintenance to be performed only when necessary. In addition, it enables the early

diagnosis of faults by applying prediction tools based on historical data, ad-hoc defined health factors, statistical inference methods, and engineering approaches.

The expansion of sensor technology has substantially increased the amount of data collected from machines and industrial processes, enabling Machine Learning (ML) to emerge as a potent tool for constructing intelligent prediction algorithms in several applications. ML algorithms can manage high-dimensional and multivariate data and identify underlying correlations within that data in complex and dynamic situations, thereby enabling potent predicting methods for PdM applications [10].

Since the efficacy of predictive maintenance also depends on selecting an appropriate machine learning approach, we will cover the different predictive maintenance models in this section.

2.1.1 Machine Learning methods

2.1.1.1 Support Vector Machines (SVM)

Support Vector Machines is a technique for supervised learning that can be applied to classification and regression applications. SVMs aim to identify the hyperplane in a high-dimensional space that most effectively divides the two groups [11]. In the case of classification, the SVM method identifies the hyperplane that splits the data into classes based on the greatest distance (also known as margin) between the closest points in each class. In regression, the SVM algorithm identifies the line best fitting the data. SVMs are a popular option for classification jobs, since they perform well on various data sets and are easy to build.

2.1.1.2 K-nearest neighbors (k-NN)

K-nearest neighbours is an instance-based, non-parametric classification [16] and regression approach. It is referred to as "non-parametric" because it makes no assumptions about the underlying data distribution, and it is referred to as "instance-based" because it does not develop a model from the data but rather generates predictions based on the similarity between new input data and the training data. When a prediction is required, the model searches the training data to identify the K training instances most similar to the input data (based on some distance metric). The prediction result is then based on the class labels or values of the K closest neighbours. K-NN is a simple and successful technique for various prediction problems, but it may be computationally costly and needs a large amount of memory to store the training data. It may also be sensitive to the distance measure used and the value of K.

2.1.1.3 Random Forest (RF)

Random forest is an algorithm used for classification and regression in machine learning. It is an ensemble model consisting of several independent decision trees that combine to create predictions [9]. Each decision tree is trained on a random sample of the data and generates predictions

based on a subset of the features. The final forecast is then derived by combining the predictions from all decision trees. By training each decision tree on a distinct subset of the data, the final model is expected to be more robust and accurate. Random forests are widely used because they can accommodate many characteristics, are very easy to train, perform well across various applications, and are reasonably easy to deploy.

2.1.1.4 Logistic Regression (LR)

Logistic regression is a statistical technique to model the relationship between a category or binary dependent variable and one or more independent variables [29]. This method is based on the logistic function, which models a binary outcome by predicting the probability of the event of interest when the dependent variable is binary. The logistic function is a nonlinear function that represents the linear relationship between independent factors and the log-odds of the binary result. This method is called Multinomial logistic regression when the dependent variable is a categorical variable with more than two levels. The logistic function is used to model the connection between the independent factors and the likelihood of each category of the categorical dependent variable in multinomial logistic regression. The model is estimated using the maximum likelihood estimation (MLE) method, which aims to identify the parameter set that maximizes the likelihood of the observed data given the assumed model.

2.1.1.5 Artificial Neural Network (ANN)

An artificial neural network (ANN) is a machine-learning model inspired by the structure and function of the human brain. It was first proposed in 1943 by Warren McCulloch and Walter Pitts [27]. An ANN is composed of hundreds of single units, artificial neurons or processing elements (PE), connected with coefficients (weights) to form the neural structure and organized in layers [1]. The input layer receives the raw data, one or more hidden layers perform the calculations, and the output layer provides the final result.

2.1.1.6 Autoencoders

Autoencoders are a class of neural network models designed to learn a compressed, low-dimensional representation of the input data, which can be used to reconstruct the original data with high fidelity [17]. In anomaly detection and predictive maintenance, autoencoders have shown significant potential. By training on operationally normal data, autoencoders can learn to reproduce the typical behaviour of a system and then utilize this information to identify abnormal behaviour [13]. Autoencoders can be used in predictive maintenance to detect early indicators of equipment wear and anticipate when a repair is required. Like with any other machine learning technique, autoencoders have their limitations. The two most significant obstacles are their sensitivity to the choice of hyperparameters and the possibility of overfitting while training on small datasets.

2.1.2 Previous Works

In the article Susto et al. [40], a multiple classifier machine learning (SVM and K-NN) approach for PdM is presented, as well as a study case for a PDM method for replacing tungsten filaments used in ion implantation, which is one of the most crucial phases in the fabrication of semiconductors. The author concludes that many classifiers enable the proposed tool to be used more naturally and clearly as a health factor indicator than other PdM techniques. The provided study case shows that the proposed tool offers superior performance to traditional PvM techniques and single SVM classifier distance-based Pdm alternatives, demonstrating superior performance compared to k-NN classifiers.

The paper Paolanti et al. [31] shows a machine learning architecture for predictive maintenance of electric motors based on a random forest technique in Azure Machine Learning Studio and evaluated in a real-world industrial scenario. Multiple sensors, machine PLCs, and communication protocols were used to capture the data. The research indicates that the proposed architecture exhibits positive behaviour for forecasting various machine states with high accuracy (95 %) based on 530731 data readings on 15 distinct machine features gathered in real-time from the tested cutting machine.

The research Gohel et al. [15] proposes designing and developing a machine learning system for the predictive maintenance of nuclear infrastructure. SVM and logistic regression techniques with parameter optimization were employed for the prediction. The research shows a result with a 95 % accuracy rate, which is considerably higher than other current PdMs for nuclear power plants utilizing different machine learning algorithms and feature types.

The paper Li et al. [21] describes a machine learning method for predictive maintenance in the railroad industry. The author describes several challenges in developing machine learning techniques in this industry, including the spatiotemporal incompatibility of the information collected through multiple detectors, the challenge of big data, where a railroad under study generates 3 terabytes of data per year, and the need to develop alarm rules in the context of industry operations. The classification challenge is solved using a customized SVM approach using large-scale data. The research concludes that the suggested solution would provide considerable business value to maintenance operations.

The paper Kanawaday and Sane [20] examines the implementation of AutoRegressive Integrated Moving Average (ARIMA) forecasting on time series data acquired from numerous sensors on a Slitting Machine to predict potential failures and quality issues, hence improving the whole manufacturing process. The study presents a system architecture that integrates ARIMA and other supervised models, which are trained with the same dataset and then stacked. For new and unobserved production cycles, the ARIMA model predicts the parameter values for the rest of the cycle, and these values are provided to the supervised model for classification. The author concludes that IoT-based machine learning will assist in overcoming severe constraints on productivity and related maintenance expenses. The supervised models may be used to extract insights from the data, and the following use of prognostics and forecasting will ensure that the manufacturing process

runs smoothly, incurs minimum maintenance costs, and minimizes product quality deterioration.

This paper Biswal and Sabareesh [7] discusses the development of a bench-top test rig intended to simulate the operating conditions of a real wind turbine and to be used for monitoring its condition in order to diagnose the onset of faults in its critical components using an ANN. The neural network with five hidden layers was trained using a feed-forward network and the technique Gradient Descent with Momentum. The results demonstrate an accuracy of 92.6 %, accurately identifying defective components from healthy ones.

The paper Huuhtanen and Jung [18] used a Convolutional neural network (CNN) to monitor solar panels' functioning by predicting the daily electrical power curve of a panel based on the power curves of adjacent panels. Concerning CNN's architecture, the paper investigated two options: two fully convolutional layers (CC) and a fully convolutional first layer with unshared convolution for the second layer. The author concludes that CNN-based approaches offer promise for PdM and notes that the performance of both architectural options may be enhanced by hyperparameter optimization.

The paper Praveenkumar et al. [32] uses an SVM for fault detection to identify failure in any of the gearbox components using vibration signals. The gearbox comprises four sets of gears with distinct speed ranges. In the experiment, the SVM classifier's accuracy in gear 1 was 96.62 %, in gear 2 it was 92.375 %, in gear 3 it was 98.75 %, and in gear 4 it was 100 %. The findings indicated that the SVM demonstrated higher classification capabilities in detecting numerous defects in the gearbox and could be employed for automated fault diagnostics.

In the paper Dos Santos et al. [12], a Random Forest and Park's Vector were utilized to identify stator winding short-circuit defects in squirrel-cage induction motors by scoring the imbalance in the current and voltage waveforms as well as the unbalance in Park's Vector for both current and voltage. The author proposes two approaches: a two-classifier approach and a single-classifier approach. The experiment demonstrated that the two-classifier approach has greater predictive power than the single-classifier approach. Both approaches can detect inter-turn short circuits using only 120 data points, potentially resulting in less computational effort.

The article Sharma and Kalra [38] proposed a machine learning (ML)-based regression model to forecast the remaining usable life of a commercial vehicle tyre based on the vehicle's and tyre's historical and current condition and performance. This study compared five machine learning algorithms: Decision Tree, Gradient Boost, LifhtGBM, KNN regression, and Random Forest. Random Forest showed the best performance, achieving an accuracy of 89.99 %, leading the author to conclude that random forest can accurately predict tyre mileage based on parameters and is a cost-effective method for predicting tyre life.

In this article Bampoula et al. [4], an LSTM Autoencoder approach is proposed for evaluating the state of a hot rolling milling machine and determining its RUL by using real-world data. In this approach, three LSTM autoencoders are trained with the temporal data related to the corresponding machine's status (high, medium, and low). The author concluded that with his approach, preventive production line stoppages and the cost of maintenance operations can be decreased. Among the constraints of this research are the necessity for several neural networks and labelled

data for each condition to identify the various states. No comparison with other machine learning methods is provided.

The article Dou et al. [13] proposes a predictive maintenance approach for a Proton Pencil Beam Scanning (PBS) system by utilizing a Long Short-Term Memory (LSTM) Autoencoder model and real-world data. The model was trained unsupervised using data from normal sessions to acquire the characteristics of regular machine operation. Anomaly was quantified through the multivariate deviation between the predicted data of the model and the measured data of the day using the Mahalanobis distance (M-score). The author concluded that the proposed model allows for highly discriminative prediction of anomalous machine events, with an Area Under the Receiver Operating Characteristic Curve (AURPC) of 0.60 and 0.82, as well as an Area Under the Receiver Operating Characteristic (AUROC) of 0.75 and 0.92 for the 2018 and 2020 datasets, respectively.

In the table 2.1, we illustrate, for each previous work about predictive maintenance analyzed in this dissertation, the machine learning model, the target equipment, the year of the article, and the respective evaluation metrics employed.

Reference	**ML Methods**	**Equipment**	**Year**	**Evaluation Metrics**
[40]	SVM, K-NN	tungsten filaments	2014	Accuracy, Precision, Recall
[31]	Random Forest	electric motors	2018	Accuracy, Precision, Recall
[15]	SVM and Logistic Regression	nuclear infrastructure	2020	Accuracy, Precision, Recall, TPR, FPR
[21]	SVM	railroad industry	2014	TPR, FPR
[20]	ARIMA	Slitting Machine	2018	Accuracy
[7]	ANN	wind turbine	2015	Accuracy
[18]	CNN	solar panels	2018	
[32]	SVM	gearbox	2014	
[12]	RF	squirrel-cage induction motors	2017	Accuracy, Recall, Specificity
[4]	LSTM autoencoders	Machine	2021	accuracy, recall, precision, specificity, F1-score
[13]	LSTM autoencoders	proton radiotherapy delivery system	2022	AUPRC , AUROC

Table 2.1: Table with the previous PdM works reviewed in this dissertation

2.2 Explainable artificial intelligence

Explainable artificial intelligence (XAI) is artificial intelligence whose predictions and behaviours can be explained to humans. Unfortunately, present prediction models are so complex that it is difficult to comprehend their behaviour and predictions. To solve this problem, XAI techniques are being developed to build more explainable models while maintaining high learning performance.

2.2.1 Model classification based on explainability

Machine learning models are commonly classified into three types based on their explainability: black-box, grey-box, and white-box models.

White-box models are models for which it is possible to explain how the model operates, makes predictions, and which variables have the most significant impact; examples include linear models, rule-based models, and decision trees. Although this model's simplicity allows us to understand its inner logic, it is also the reason why it cannot achieve the same level of precision as black-box models.[23]

Black-box models are the antithesis of white-box models. Due to the effort to provide more precise predictions, the models become too complicated to comprehend the decision-making process, and the observer can only see the input and output. By nature, ensemble models and all types of deep learning are black-box models.

Grey-box models attempt to capitalize on black-box model's precision while incorporating interpretability. Although the model's operation is not entirely transparent, the model does explain the relationship between input data and outcome.[8]

2.2.2 Intrinsic and post-hoc

The term "intrinsic interpretability" describes a machine learning model with a clear structure, which is accomplished by restricting the complexity of the model. This characteristic is often associated with white-box models, such as decision trees and sparse linear models, which include components, such as paths and rules, that can be directly examined to comprehend the model's prediction and provide traceability and transparency[33]. An approach was proposed in the paper Islam et al. [19] that reduces the complexity of the black-box model with little or no compromise in performance. This is accomplished by exchanging features of a model that are difficult to interpret with easily interpretable features induced from domain knowledge.

In contrast to the intrinsic interpretability that must be applied during the pre-modelling or modelling phases, post hoc explanations provide interpretability after a model has already been developed. Besides the possibility that post hoc interpretations may be misleading [22], they are gaining popularity since they provide a means to achieve prediction accuracy and interpretability.

2.2.3 Global and local methods

Local explainability methods deliver detailed explanations for why the model reached a specific outcome [39]. This approach helps determine why a specific outcome failed or explains the decision to the impacted party. Global explainability methods explain the model's behaviour by showing which input parameters influence the overall prediction accuracy [3]. However, correlations in the input features may jeopardize explainability.

2.2.4 Model-specific and model-agnostic

When a technique of interpretation is model-agnostic, the method can be used to interpret any machine learning model [46]. This technique cannot access the model's underlying architecture, so it concentrates on analyzing pairs of input and output attributes. Therefore, it must be applied after the model has been trained (post hoc).

On the other hand, model-specific interpretation techniques can only be used with particular models and have access to the model's structure and learning processes [33]. The interpretation of regression weights in a linear model, a feature of his architecture, is a model-specific method that cannot be applied to other prediction models, such as neural networks.

2.2.5 Explanation Models

2.2.5.1 Partial dependence plots (PDP)

The PDP is a model-agnostic global method of explanation that demonstrates the impact of one or two parameters on the predicted output of a machine learning model [14] and will reveal if the relationship between the outcome and a feature is linear, monotonic, or more complicated [28]. This model has several advantages: it is simple to implement, and the interpretation is straightforward when the variable inputs are uncorrelated. Therefore, the plot can demonstrate how the average prediction changes as the features are altered. The explanation is intuitive so that laypeople can comprehend it. Unfortunately, this method has a few drawbacks: Not displaying the feature distribution might be deceptive since it is possible to overinterpret areas with little data [28]. The independence assumption is the most problematic aspect of PDP since altering values when the feature is correlated leads to unreal scenarios, risking the accuracy of the interpretation. Since PDP only displays the average marginal effects, heterogeneous effects may be concealed. Because the visualization is limited to 1D or 2D, certain relevant information about the model may be absent from the plot [41]

2.2.5.2 SHAP (SHapley Additive exPlanations)

Shapley additive explanations is a model-agnostic technique proposed by Lundberg and Lee in 2017 [24] in which the core idea is to compute the Shapley values for each data variable interpreted, where each Shapley value quantifies the effect that the associated feature has on the prediction. The Shapley values SHAP uses are the average marginal contribution of an instance of a feature over all potential coalitions. Lloyd Shapley introduced this approach in 1951 [37]. Due to the usage of Shap values, the SHAP has several benefits, including his forecast being evenly distributed throughout the feature values and a solid theoretical base in game theory[28]. However, it also has some disadvantages, such as the high cost of computing; as the number of features increases, more combinations must be calculated.

2.2.5.3 Local interpretable model-agnostic explanations (LIME)

LIME, a local model-agnostic proposed by Ribeiro et al. [34], is an algorithm capable of accurately explaining the predictions of any classifier or regressor by approximating it locally with an interpretable model. As is the case with other types of surrogate models [28], the first step is to pick an instance for which you want an explanation of the model's prediction. Next, it disrupts the dataset by performing variations on the data. After that, it weighs the data points based on how close they are to the instance. Finally, train a weighted, interpretable model with the disrupted dataset to explain the prediction. The Lime has the benefits of being incredibly simple to implement, containing a metric that indicates how well the interpretable model explains the black box prediction and applying to tabular data, text, and images.

2.2.5.4 Adaptive Model Rules (AMRules)

Adaptive Model Rules, introduced in Almeida et al. [2], is the first rule-learning algorithm for data stream regression problems. The antecedent of a rule in AMRules is a conjunction of conditions on attribute values, while its consequent is a linear combination of the attributes. Each rule employs a Page-Hinkley test to detect changes in the data generation process and reacts to changes by pruning the rule set. AMRules allow global and local explanations Ribeiro et al. [35]. Globally, it provides a set of learned rules to elucidate the conditions for high predicted values. Locally, we gain an understanding of the rules triggered by specific input.

2.2.6 Previous Works

In the article Szepannek and Lübke [41], The author begins by highlighting the impacts that a failed prediction in the forensics field can have on people's lives and then proposes the PDP method to explain the prediction model random forest, which is trained on a popular real-world data set: the glass identification database, to predict the type of glass based on chemical analysis. Using the PDP, the author found significant nonlinearities and explained which parameters affect the outcome most. However, he concluded that the plots only partially explain the model. To quantify how much of the model's prediction is visualized by PDP, the author employs the measure of explainability, measured by the differences between the partial dependence function and the model's prediction.

The research Barredo-Arrieta et al. [5] uses the model-agnostic SHAP to reveal the knowledge collected by Random Forest and Recurrent Neural Networks for predicting real-world traffic. According to the author, the capability to recover knowledge gathered by traffic forecasting models may aid in comprehending how to enhance their design and extract insights. Using SHAP, the author's study revealed that the two models weighted their predictors differently, indicating that they had acquired different knowledge. The article concludes that explainability methods provide researchers with two crucial capabilities: the ability to verify if the model's acquired knowledge corresponds to intuition and the capacity to predict future outcomes, and the ability to study deeper what the model focuses on.

The article Islam et al. [19] begins by discussing the significance of explaining black box models and drawing attention to the European Union-approved "right of explanation" law. In their introduction, the authors assert that the post hoc idea of interpretability needs to be more transparent and may be deceptive since it explains after a choice has been made. Instead of a post hoc explanation strategy, this study focuses on interpretability in the pre-modelling phase, employing domain knowledge in the context of bankruptcy prediction. In the authors' method, they replace the difficult-to-understand attributes of a model with simple-to-understand attributes induced by domain knowledge. This was accomplished using Apriori, a frequent pattern mining algorithm proposed by Agrawal, to find frequent feature sets used in different bankruptcy literature. They then related the frequent feature set to popular financial concepts of credit to generalize the features, allowing domain knowledge to be infused to improve the model's explainability. The "Freddie Mac dataset"[25] was used to train five prediction models (artificial neural networks, support vector machines, random forests, extra trees, and gradient boosting) in the experiment. The author concludes that his technique permits a better explanation of "black box" models without sacrificing performance significantly.

The article Torcianti and Matzka [43] predicted machine failure using a complex model classifier, a bagged tree ensemble trained with the dataset of predictive maintenance supplied by Matzka [26]. In order to provide a credible explanation for the classification result, the author compares three explanation methods: LIME, Normalized Feature Deviation, and Explainable Decision Trees. In order to provide a reasonable explanation for the classification result, the author compares three explanation methods, namely LIME, Normalized Feature Deviation, and Explainable Decision Trees, and concludes that LIME has the highest overall explanatory quality. The author also underlines the major unresolved issue with LIME, the instability of explanations, and says that SHAP is preferable.

The article Terziyan and Vitko [42] explains the significance of artificial intelligence in Industry 4.0, its benefits, and the necessity of explaining machine learning models without compromising their efficacy. To address this issue, the author proposed representing the black box model as an explainable decision tree and transforming it into Semantic Web Rule Language (SWRL), a standard format for expressing rules and logic. The author concludes that it is possible to retrain deep learning results as decision trees without access to the original training data and essential loss of accuracy and shows that the retrained decision tree models can be represented as SWRL rules. This study was restricted to classification problems and only considered models derived from numerical data, which is a limitation of the approach.

The article Amiri et al. [3] uses an artificial neural network as a transportation energy model, trained with Household Travel Survey data, to predict household transportation energy and uses Lime to generate explanations, which the author considers essential for ensuring trust and providing important information. Using LIME for individual explanations and a Submodular Pick (SP-LIME) for explaining the prediction model, the article concludes that the results are promising, providing useful insights for and enabling machine learning experts to better comprehend the intrinsic data characteristics and apply feature engineering techniques.

$$MDMC = \frac{1}{n}\sum_{i=1}^{n} D = \frac{1}{n}\sum_{i=1}^{n} f(M - M^{*})$$
$$= \frac{1}{n}\sum_{i=1}^{n}\left[(R2_0 - R2_i) + (MSE_i - MSE_0) + (MAE_i - MAE_0)\right]$$

Figure 2.1: MDMC equation

In this article Panigutti et al. [30], Doctor XAI, the first agnostic explanation technique applicable to any black-box classifier dealing with sequential, multi-labelled, and ontology-linked data, is proposed to explain the Doctor AI, a Recurrent Neural Network (RNN) multi-label classifier that uses a patient's clinical history to predict the next visit. The approach outlined in the article implies generating a set of synthetic instances similar to the instance whose black-box decision we need to explain, training an interpretable classifier on this neighbourhood, and then extracting a rule-based explanation from it. This paper focused on the medical domain, but since the technique is agnostic, its potential applications span a variety of scenarios in which we can identify sequences of events linked to ontology concepts. Future research by the author will examine other types of synthetic neighbour generation for sequential data and assess the effect of the random components of the synthetic neighbour generation procedure on the quality of explanations. The author also claims that he could explain black-box regressors with a simple extension.

In this article, Zhang et al. [47], the Mean Degree of Metrics Change (MDMC) XAI evaluation framework is established to compare and select XAI methods more suitable for the model. The authors employ three prediction models in the experiment: ANN, representing neural network models; LightGBM and Random Forest, representing ensemble models. SHAP and LIME were utilized to explain the prediction models, which were afterwards evaluated and compared using MDMC. The established XAI evaluation framework is predicated on a permissive assumption: deleting or modifying the large contribution features in the dataset will significantly reduce the accuracy of the model's predictions. The MDMC can be calculated using the equation in Figure 2.1. The larger the result, the more likely the prediction model has substantially changed the dataset, demonstrating XAI's efficacy. After the experiment, the results indicate, according to the MDMC, that LIME performs better in the ANN and random forest models, while SHAP performs better in the LightGBM model.

This article van der Waa et al. [44] evaluates the effects of rule-based and example-based contrastive explanations on system comprehension, persuasion, and task performance. This evaluation consisted of two experiments involving 45 participants. The experiments showed that both methods enabled participants to correctly identify the situational factor that played a decisive role in a system's advice and increased the frequency with which they followed it. However, neither method enabled participants to correctly predict the system's advice in novel situations or improve task performance, and the experiment demonstrates that more than one method is required to improve a user's understanding or task performance.

In this article Sahoo et al. [36], to explain the black-box features of data-driven ML models used to control power electronic converters, the author proposes an explainability framework that begins by calculating the conditional entropy of any specific input that may have on the predicted outcome, then extracts its average influence by returning conditional prediction weights, and finally removes the corrupt data in the training dataset by removing the outliers. In conclusion, the author enumerates several advantages, such as reducing the dimensionality and size of the training dataset by removing abnormal setpoints, increasing data quality, and the offline diagnosis platform for the predicted values.

In this article [35], an LSTM-Autoencoder is proposed to detect anomalies in Volvo city buses. Additionally, the algorithm AMRules is used, which employs the high reconstruction error values obtained from the LSTM-Autoencoder to generate both global and local explanations. The global explanation takes the form of a rule-set, while the local explanation identifies the specific rule that triggered the anomaly for a given instance. This article additionally presents two distinct sampling strategies, namely the ChebyUS and the ChebyOS, which aim to enhance the learning of rules related to rare cases. We can conclude that the results show great success, clearly illustrating that it is possible to identify and interpret anomalies in real time, providing compact explanations.

In the table 2.1, we illustrate, for each previous work about explainable machine learning models analyzed in this dissertation, the XAI method, the target machine learning model, and the year of the article.

Reference	**XAI Methods**	**ML Methods**	**Year**
[41]	PDP	RF	2022
[5]	SHAP	RF, RNN	2019
[19]	Domain Knowledge (Apriori)	ANN, SVM, RF, Extra Trees, GB	2019
[43]	LIME	Bagged Tree Ensemble	2021
[42]	SWRL Rules	ANN	2022
[3]	LIME	ANN	2021
[30]	Doctor XAI	Doctor AI (RNN)	2020
[47]	SHAP, LIME	ANN, LGBM, RF	2021
[44]	Rule-based explanations		2021
[36]	Proposed Model	ANN	2021
[35]	AMRules	LSTM-Autoencoder	2022

Table 2.2: Table with the previous XAI works reviewed in this dissertation

2.3 Summary

This chapter examined several maintenance strategies, with PdM being the most effective since it avoids failures and only performs maintenance when necessary, thereby reducing costs. We also analyzed several machine learning approaches and their performance in previous works, finding that they can be used successfully in the design of PdM applications. In this chapter, we also examined the explainability of artificial intelligence, where models can be categorized as white, grey,

or black box models based on their interpretability. We also discuss the scope of the explainability, whether it is local, where we attempt to explain a single model decision, or global, where we attempt to explain the entire model. We illustrate the distinctions between model-specific, model-agnostic, intrinsic, and post-hoc. In addition, we present various XAI approaches, such as Lime and SHAP, along with their respective performances in previous research. Our dissertation distinguishes itself from previous research by placing an important focus on enhancing the explainability of the LSTM Autoencoder, used as an unsupervised anomaly detection method within the domain of predictive maintenance, by employing and comparing two distinct explainability models, the renowned SHAP and the AMRules, which represents a novel approach to explainability, described in the article Ribeiro et al. [35].

In the following chapter, we conduct an exploratory analysis of our dataset, construct our proposed model for predictive maintenance, and explore the model's explainability.

Chapter 3

Matherials and Methods

This chapter describes the methods and strategies utilized in our research. In the first section, we introduce the datasets used in this research, namely MetroPT and NASA. In the Data Analysis section, we discuss a series of issues related to the MetroPT dataset. In the section Anomalies Methods, we introduce our LSTM Autoencoder by describing its architecture, training procedure, and implementation in order to facilitate future replication. In addition, we describe the evaluation procedure for our model, for which we offer two distinct approaches. Finally, we present the two employed XAI techniques for knowledge extraction and explanation of our black box LSTM Autoencoder's predictions.

3.1 Data Description

3.1.1 MetroPT

This research uses the real-world dataset MetroPT, collected in 2022 in Porto, Portugal. This dataset was compiled due to a Predictive Maintenance project with a metropolitan public transportation system. The principal aim of this dataset is to facilitate the development and evaluation of machine-learning techniques for anomaly identification and failure prediction. Each row represents a second of sensor activity that includes a variety of analog sensor signals, such as pressure, temperature, and current consumption, as well as digital signals and GPS data (latitude, longitude, and speed). In this study, we will only focus on the analog sensors.

The first version of the dataset is composed of 10979545 rows and has three failures, as we can see in Table 3.1. The first failure was an air leak in a pipe that feeds multiple clients on the systems, such as the breaks, suspension, etc. The second was an air leak on the air dryer caused by a malfunction of the pneumatic pilot valve that opens the drain pipes when the compressor is operating, and the third was an oil leak on the compressor that severely damaged the compressor's motor.

The second version of the MetroPT dataset comprises 7940116 rows and contains two failures, as we can see in Table 3.2. The first is an air leak with a duration of approximately two hours, and the second is an oil leak with a duration of approximately three days.

Table 3.1: MetroPT V1 Failures

Nr.	Start	End	Type
1	28/02/2022 21:53	01/03/2022 02:00	Air Leak
2	23/03/2022 14:54	23/03/2022 15:24	Air Leak
3	30/05/2022 12:00	02/06/2022 06:18	Oil Leak

Table 3.2: MetroPT V2 Failures

Nr.	Start	End	Failure
1	2022-06-04 10:19:24.300	2022-06-04 14:22:39.188	Air Leak
2	2022-07-11 10:10:18.948	2022-07-14 10:22:08.046	Oil Leak

3.1.2 NASA Bearing

The NASA Bearing Dataset is a publicly accessible dataset used extensively for diagnostics and predictive maintenance of rotating machinery. It was collected by the Prognostics Center of Excellence at NASA Ames Research Center. The dataset contains sensor readings from four accelerometers installed on the bearing housing of a NASA turbofan engine. The engine was subjected to various operating conditions and faults to simulate real-world scenarios.

3.2 Data Analysis

In this section, we used Power BI as a visualization tool to display the analog sensor data acquired by the MetroPT dataset. The primary objective was to gain insight into the system's operation-phase performance and behavior. Nonetheless, it is essential to acknowledge a limitation encountered during data analysis. We encountered challenges when attempting to process the data at a granularity of seconds. Due to this constraint, we aggregated the data into daily intervals, displaying the key statistical metrics average, maximum, and minimum values for each day. Despite this approach providing a comprehensive overview of the sensor readings, a finer granularity analysis could have provided more information about the system's dynamics during specific time intervals.

Figures 3.1 and 3.2 illustrate the sensors in the first version of the MetroPT dataset, where we identify two failures. The first and most critical problem is the severe outliers that affect all sensors simultaneously but are not considered failures by the dataset. A human eye can easily detect these outliers. The second issue is with the DV_pressure sensor; before March 29, the average daily maximum values fluctuated between 8.33 and 6.32. However, after this date, the average daily maximum values dropped to an astonishingly low 0.70, a behaviour that we cannot explain. Since autoencoders learn from normal data, it is important to mention these issues as they may penalise the predictive model's performance.

Upon analysis of the second iteration of the MetroPT dataset, we encountered different issues. All sensors appeared to operate within the expected parameters. Although the dataset contains outliers, as depicted in figures 3.3 and 3.4, it is noteworthy that the most pronounced outliers align precisely with the documented failures within the dataset. This alignment is consistent with the

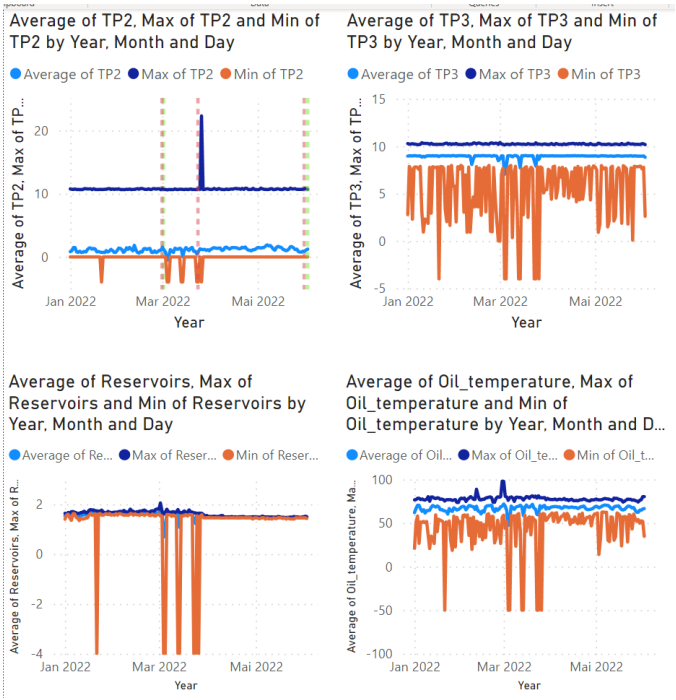

Figure 3.1: MetroPT V1: TP2, TP3, Reservoirs, Oil

inherent nature of anomalies, underscoring the dataset's veracity in capturing exceptional occurrences.

3.3 Anomalies Methods

We propose the model LSTM (Long Short-Term Memory) Autoencoder to detect anomalies. The Autoencoders are trained on normal operating conditions and learn how to reconstruct the input data in this normal pattern. Therefore, in a faulty case where the data deviates from the normal pattern, the reconstruction error increases, indicating the presence of anomalies. Implementing the LSTM cells in an Autoencoder architecture enables us to handle long-term dependencies in the sequential data, which is essential for capturing temporal patterns. Since the LSTM autoencoder is a self-supervised learning technique that focuses on normal data, it does not require a labelled fault during training, which is a significant advantage for anomaly detection problems because we do not need to represent in the dataset all possible anomalies that the model will need to detect.

3.3.1 Model Achitecture

Our proposed architecture for the LSTM Autoencoder, represented in the figure 3.5, begins with an input layer containing eight neurons that receive a data sequence. The succeeding layers, L1 and L2, are LSTM layers containing six and four neurons, respectively. An intermediary layer, L3, the repeat vector layer, prepares the data for the subsequent LSTM layers responsible for the input sequence's reconstruction. LSTM layers L4 and L5 comprise four and six neurons,

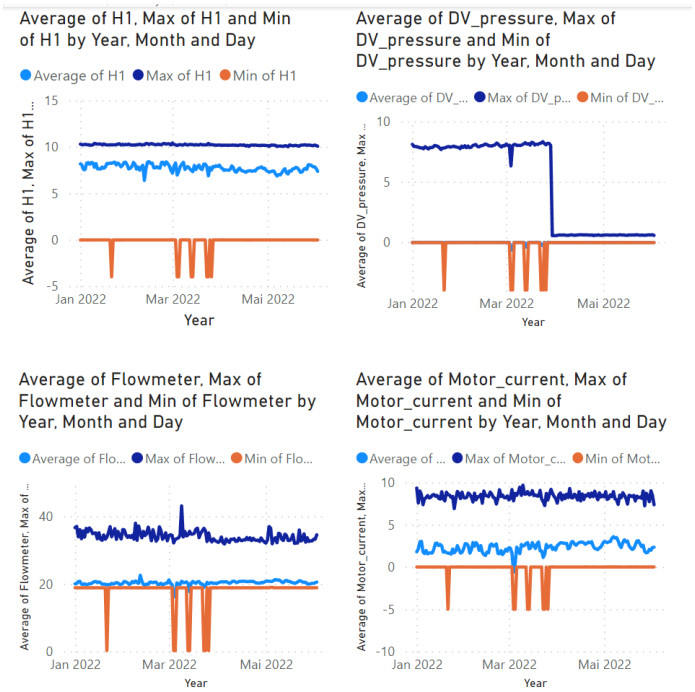

Figure 3.2: MetroPT V1: H1, DV_Pressure, Flowmeter, Motor Current

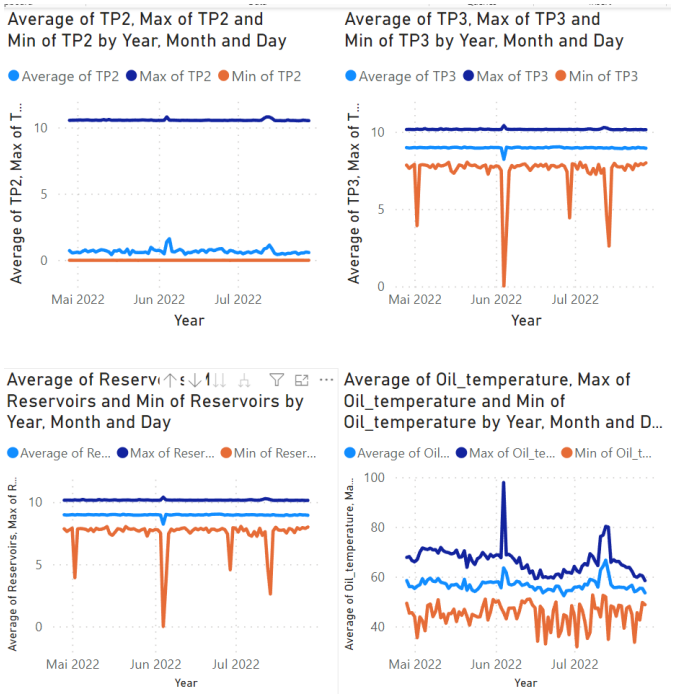

Figure 3.3: MetroPT V2: TP2, TP3, Reservoirs, Oil

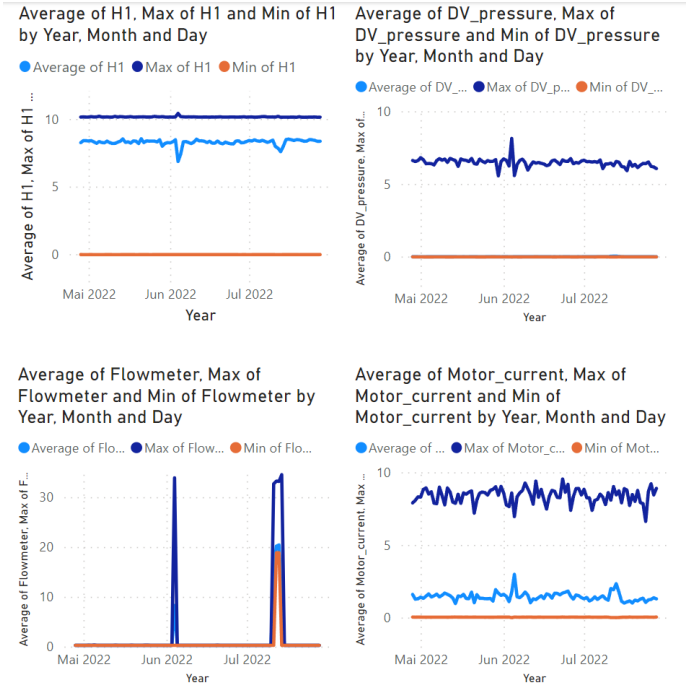

Figure 3.4: MetroPT V2: H1, DV_Pressure, Flowmeter, Motor Current

respectively. The final layer forms a TimeDistributed Dense layer, orchestrating the generation of the reconstructed sequence via a linear activation function. Notably, the LSTM layers L1, L2, L4, and L5 adopt the hyperbolic tangent (tanh) activation function. A Batch Normalisation layer is applied to these layers to normalise the activations and accelerate the training process. After that, the dropout layer is applied to randomly deactivate a fraction of neurons with a rate of 0.2 to prevent overfitting. Our LSTM Autoencoder architecture for the NASA dataset follows a similar blueprint but has different neuron counts. It begins with an input layer containing four neurons, followed by L1 with three neurons, L2, L3, and L4 with two neurons each, and L5 with three neurons. The final layer remains consistent, with four neurons in the output layer.

3.3.2 Training

In this section, we start by discussing the training of our MetroPT model. The initial step consisted of dividing the dataset into training and testing subsets. However, we needed more resources to use the entire dataset. Therefore, experiments were conducted to determine the most suitable training data required. Through our experiments, we discovered a curious pattern. Training the model with data from an entire month yielded less accurate predictions than training it with a week. At this stage in the development process, our access to the MetroPT dataset was restricted to the first version, which exhibited some sensor inconsistencies. Consequently, these sensor issues contribute to the difference between training with a month's worth of data and just a week's data. To identify the seven-day training dataset from the first version of the MetroPT dataset, certain factors highlighted in the data analysis section needed careful attention, which imposed a few

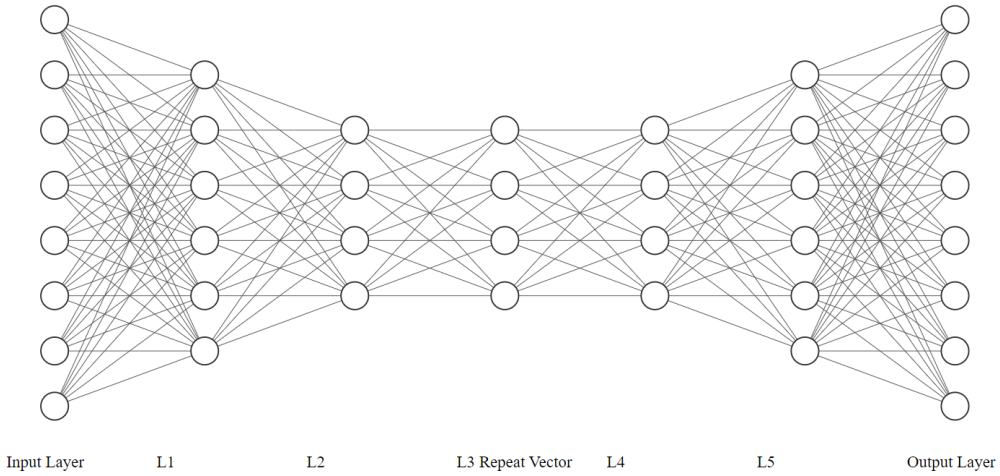

Figure 3.5: Proposed LSTM Autoencoder Achitecture

constraints on this selection process. After March 29, the DV_Pressure sensor was excluded due to its peculiar behaviour. In addition, March, which was characterized by the presence of scattered outliers, was considered irregular and therefore disregarded for training selection. Accounting to these factors, the focus narrowed down to January and February. However, it is important to note that January 21 emerged as an outlier. Consequently, the specific date was disregarded. Experiments were conducted to determine the best training week. However, none had significantly improved performance, so we decided to use the first seven days of the dataset for training and the following seven for testing. Identifying the seven-day training from the second version of the MetroPT dataset was easier because we didn't encounter the same issues as with the first version. We only needed to avoid the first small outlier on May 2. The experiments to determine the optimal seven days for training delivered the same result as the original, but none significantly improved performance. Considering this, we used the seven days between May 3 and May 10 as training data. During the training of our NASA model, we did not encounter the same constraints present in the MetroPT dataset. Consequently, the entire dataset was available for the training process. The machinery begins in optimal condition in this data set and deteriorates progressively over time. To divide the dataset into training and testing subsets, 45 % of the data was allocated for training, while the remaining 55 % was reserved for testing.

3.3.3 Implementation

This section will describe how the model's training was conducted in detail. Initially, we imported the datasets into a pandas data frame and designated the timestamp column as the index. Subsequently, we divided the dataset into two sets: training and testing. In the case of the MetroPT dataset, the columns corresponding to digital sensors and GPS were also excluded. Data scaling was the next step after establishing the training and test data subsets. We utilised the MinMaxScaler from the Sklearn library to conduct the normalisation. The Keras library was used to construct the LSTM autoencoder architecture described in the model architecture section. To compile

the model, we used the Adam Optimizer with a learning rate of 0.1 % and the mean absolute error (MAE) as the loss function. For training the MetroPT model, we used 25 epochs and a batch size of 300, corresponding to 5 minutes of sensor data, whereas for the NASA model, we used 100 epochs and a batch size of 10. The LSTM autoencoder exclusively reconstructs the initial input data, necessitating the computation of the reconstruction error. We utilise the Numpy library to calculate the mean absolute error between the autoencoder's predicted values and the actual data. When the deviation between predicted and actual values exceeds the predetermined threshold, we can interpret it as an anomaly identified by the model.

3.3.4 Evaluation

For evaluating the MetroPT model's performance, we employed two distinct approaches. The first approach entails an isolated evaluation of each second, so the metrics are computed based on the model's predictions for each instance. The second approach, recommended by the creator of the dataset in Veloso et al. [45], focuses on evaluating the alignment between predictions and actual data through an overlap analysis. This approach takes failures into account rather than second-by-second performance.

To evaluate our model for predicting anomalies, we relied on a selection of important performance metrics. These metrics are essential for quantifying the model's capacity to capture underlying patterns accurately, detect anomalies, and provide a comprehensive view of its efficacy. The primary metric we considered is precision, defined by the formula:

$$Precision = TP/(TP+FP) \tag{3.1}$$

Which measures the proportion of correctly predicted anomalies (true positives) relative to all predicted anomalies. A higher precision value indicates a lower rate of false positives, reflecting that the model can detect anomalies without triggering unnecessary alarms. Additionally, we incorporated the recall metric, defined by the formula:

$$Recall = TP/(TP+FN) \tag{3.2}$$

Which indicates the model's ability to identify a greater proportion of actual anomalies. A greater recall value demonstrates the model's efficacy in detecting anomalies. The F1-score, our third metric defined by the formula:

$$F1Score = ((Precision * Recall)/(Precision + Recall)) * 2 \tag{3.3}$$

Balances precision and recall by calculating the harmonic mean of these two metrics. The F1 score provides a consolidated metric that summarises their interaction.

Due to the need for more information about the precise onset of equipment failure within the NASA dataset, it is not possible to evaluate the performance of the NASA model using the previously mentioned metrics.

3.3.5 Model Explainability

In this section, we talk about the explainability of the LSTM autoencoder, where our objective is to gain insights into the learned features and identify which aspects of the input data contribute most to the reconstruction process. Understanding these features can aid in better understanding fault characteristics and assist domain experts in diagnosing machinery faults. To obtain these insights, we employed the SHAP explainability model and AMRules.

In the training process of our SHAP explainability model, we opted to use the Kernel Explainer provided by the SHAP library. This model needs to explore and evaluate many possible feature interactions. Each feature's contribution to predictions must be computed for all possible permutations, which can result in a substantial computational workload. However, it is essential to note that our model training resources are limited. Consequently, we were only able to utilise part of the dataset. Therefore, we were forced to work with a subset of 2400 data instances,the equivalent of 40 minutes of sensor activity, to accommodate these constraints. This strategy permitted us to continue the training process despite our limited resources. To create the AMRules explainability model, we utilised the AMRules class provided by the River framework. During the training phase, we employed the ChebyshevOverSampler class from the same library, a method that involves oversampling, to enhance the model's capacity for capturing extreme instances.

3.4 Summary

In this chapter, we presented the fundamental aspects of our research. We began by introducing the datasets utilised for this dissertation, mainly the MetroPT dataset. Subsequently, we conducted a comprehensive data analysis of the MetroPT dataset, uncovering inherent issues that could affect the performance of our model.

Much of this chapter was devoted to introducing our proposed LSTM model. We illustrated its architecture, discussed the training procedure, and underlined a few of the inherent limitations of this training procedure. In addition, we looked into the technical implementation of our proposed model.

We then discussed the evaluation strategies and metrics to assess the model's capabilities. These strategies enable us to evaluate the model's efficacy, providing valuable insight into its predictive ability.

Finally, we covered the model's interpretability, employing the SHAP and AMRules methodologies to make the model's decision-making process more transparent. Our goal was to decipher the complex underpinnings of our LSTM model, comprehend the fault characteristics, and aid subject-matter specialists in identifying mechanical problems.

Chapter 4

Results and Discussion

This chapter presents an analysis of the research findings. It exhibits the outcomes achieved through the use of our proposed LSTM Autoencoder on the MetroPT and NASA datasets in order to address our research questions:

1. Is it possible to detect anomalies using an unsupervised method?
2. Can the predictions of a black box model be explained?

This chapter will address distinct sections for each of our research questions. Each section will end with a dedicated subsection to discuss our findings to respond to the question.

4.1 Is it possible to detect anomalies using an unsupervised method ?

In order to address our initial research inquiry regarding the feasibility of detecting anomalies using an unsupervised method, we will present the results of our LSTM Autoencoder in this section. We start by examining the model's training process. Subsequently, we assess the reconstruction error, which provides insights into the model's ability to replicate the input data faithfully. Following this, we employ the performance measures discussed in the previous section to assess the efficacy of utilising reconstruction error to detect anomalies. Ultimately, we discuss how the obtained results address the research question.

4.1.1 Model training

First, we analyse the training performance of our LSTM Autoencoder by plotting the model's training loss curve throughout epochs. The graph's horizontal axis represents the number of training epochs, while the vertical axis corresponds to the loss function's magnitude, the mean absolute

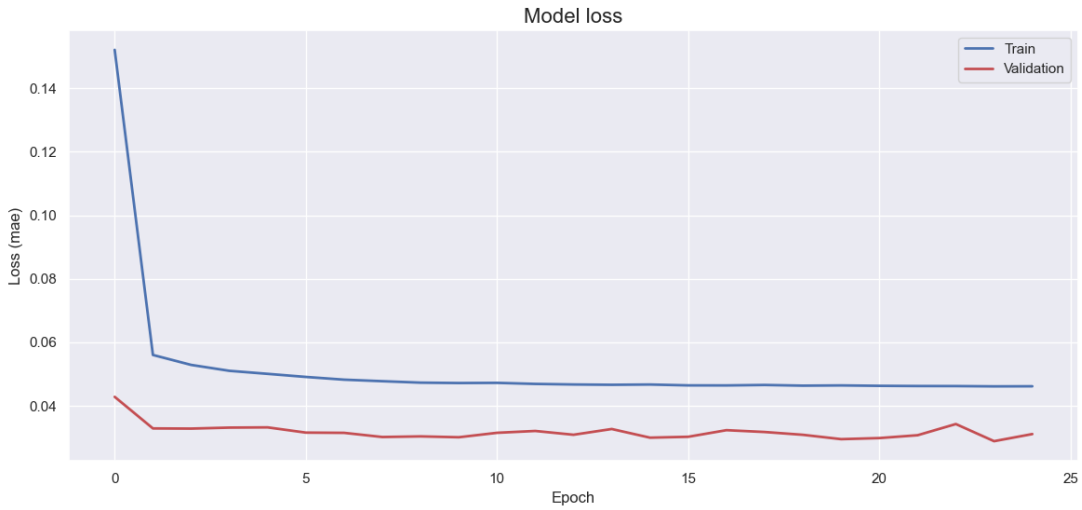

Figure 4.1: Loss function MetroPT v1

error. As shown in Figures 4.1 and 4.2, in the initial phases, the loss decreases rapidly in both figures, indicating that the models are effectively adapting to the training dataset, meaning that the model is effectively decreasing the discrepancy between its predictions and the actual target values. After the third epoch, the training loss curve exhibits a smoother decline, forming an almost straight trajectory, which suggests stabilising the model's learning process, where it achieves a certain level of data pattern comprehension. In figure 4.1, the validation loss curve consistently remains below the training loss curve, a positive sign indicating that the model can generalise effectively to unseen data. Figure 4.2 illustrates a similar trend in which the validation loss curve generally resides below the training loss curve. However, it is important to note that the validation loss curve exhibits minor fluctuations in later epochs. These fluctuations are indicators of the model's ongoing learning as it adapts to the data's complexities.

In the training of the LSTM autoencoder for the NASA dataset, as shown in figure 4.3, the loss function decreases rapidly in the first twenty epochs, indicating that the models are effectively adapting to the training dataset. After the 20th epoch, the loss function decreases gradually, indicating that the model's learning process has stabilised around the 0.10 mae error. The validation loss curve consistently remains below the training loss curve, indicating that the model can generalise effectively to new data.

4.1.2 Reconstruction error

To assess the reconstruction error of the models, we provide a visual representation of their predictive precision throughout the observation period. These graphs illustrate the mean absolute error on the y-axis, highlighting the difference between predicted values and actual data points. The x-axis represents timestamps and provides a temporal context for the model's efficacy over time. Figure 4.4, presents the model's reconstruction error based on the initial version of the dataset,

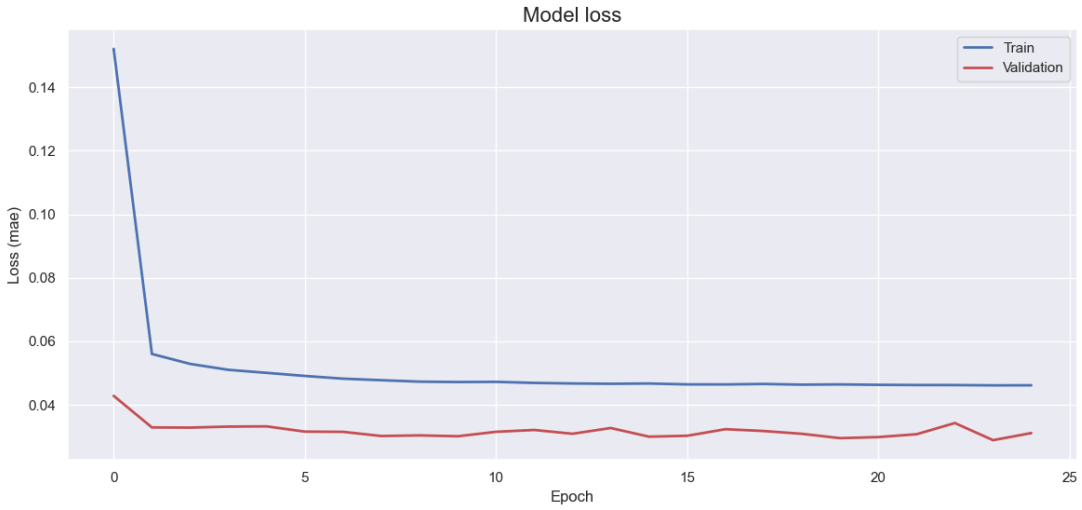

Figure 4.2: Loss function MetroPT v2

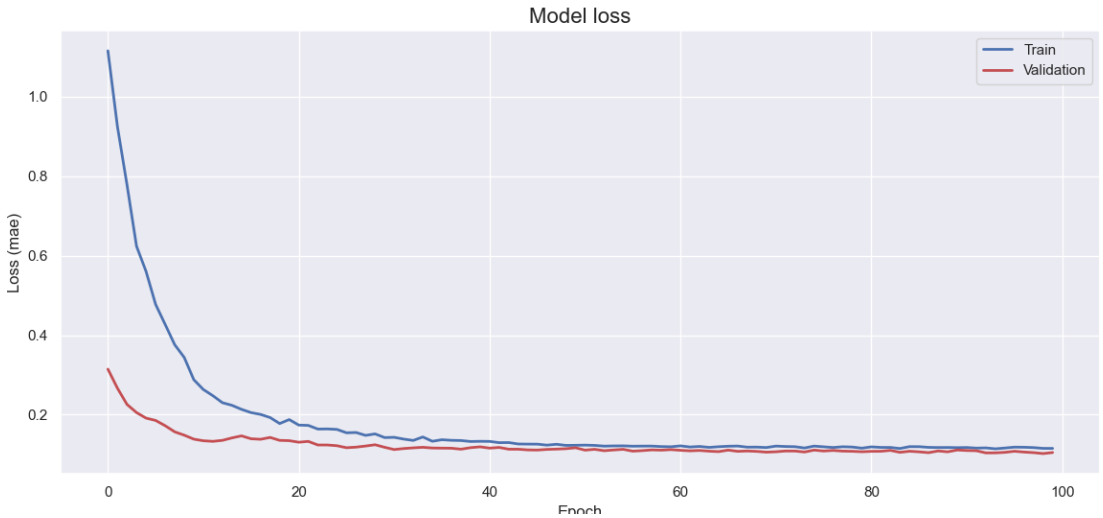

Figure 4.3: Loss function MetroPT v2

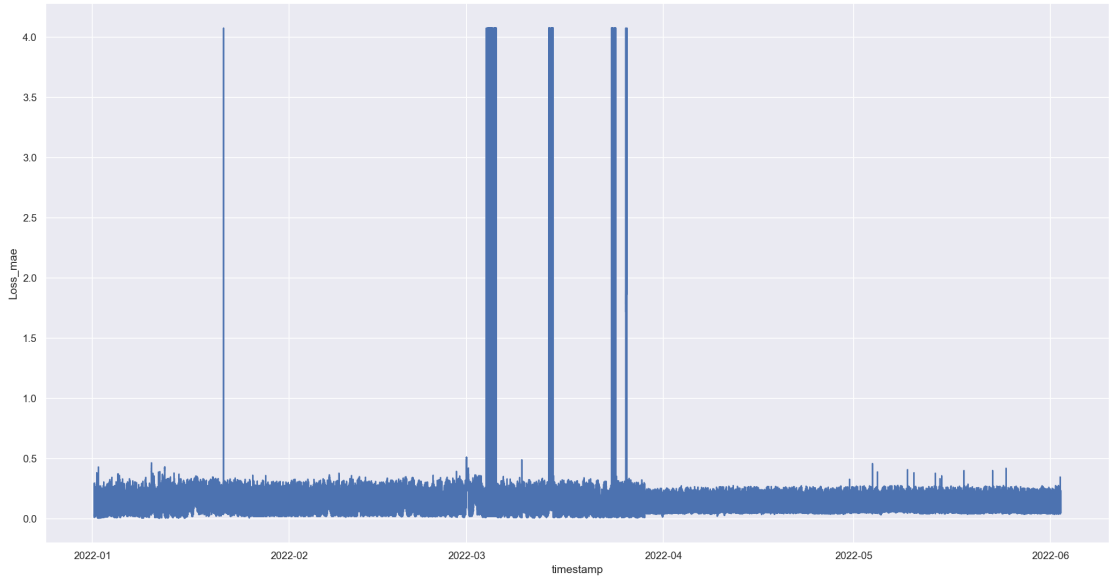

Figure 4.4: Reconstruction Error MetroPTV1 with outliers

retaining the presence of anomalies as discussed in the section on Data Analysis. A closer examination of the plot reveals that the model can identify these anomalies, as evidenced by the MAE value of 4. Despite this capability, a problem arises in that the identified outliers are not the anomalies recognised by the dataset. Moving on to Figure 4.5, we examine the model's reconstruction error within the dataset's same version but without outliers. This modification reveals a more distinct representation of the MAE fluctuations across the temporal dimension. Despite this level of detail, the anomalies defined by the dataset remain difficult to detect. Figure 4.6, which illustrates the reconstruction error of the model using the second version of the dataset, reveals the model's proficiency in detecting anomalies. Notably, the two anomalies identified by the dataset are clearly reflected in the reconstruction error plot, where the MAE error reaches a value of 60. Figure 4.7, which displays the NASA model's reconstruction error, clearly illustrates how the MAE increases over time. The MAE begins at approximately 0 and then gradually increases over time. In its later phases, it accelerates significantly, eventually exceeding a significant error value of 30. This trend is consistent with the nature of the NASA dataset, in which machinery progressively deteriorates over time.

4.1.3 Evaluation Metrics

In the first version of the MetroPT dataset utilising the first approach, our model demonstrates an approximate precision value of 0.467. This indicates that a significant portion of the anomalies detected by the model are incorrect predictions, resulting in a higher rate of false positives. The recall score of 0.023 indicates that the model is ineffective in identifying true anomalies, which reveals that it is unsuccessful at identifying actual machine faults. The F1-score is approximately 0.043, and his low score confirms the model's poor performance in detecting anomalies, producing

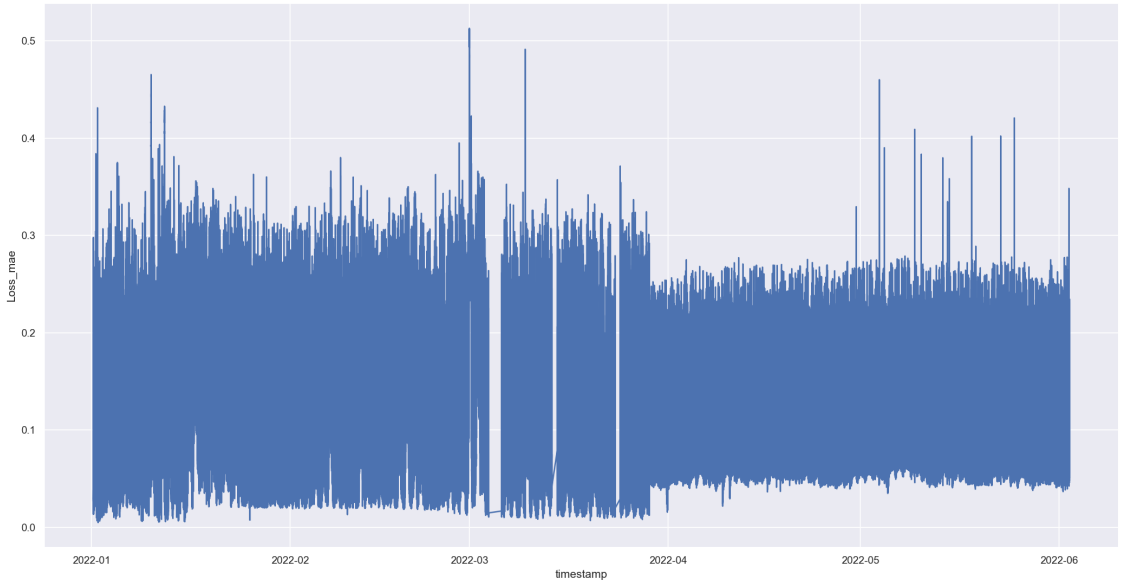

Figure 4.5: Reconstruction Error MetroPTV1 without outliers

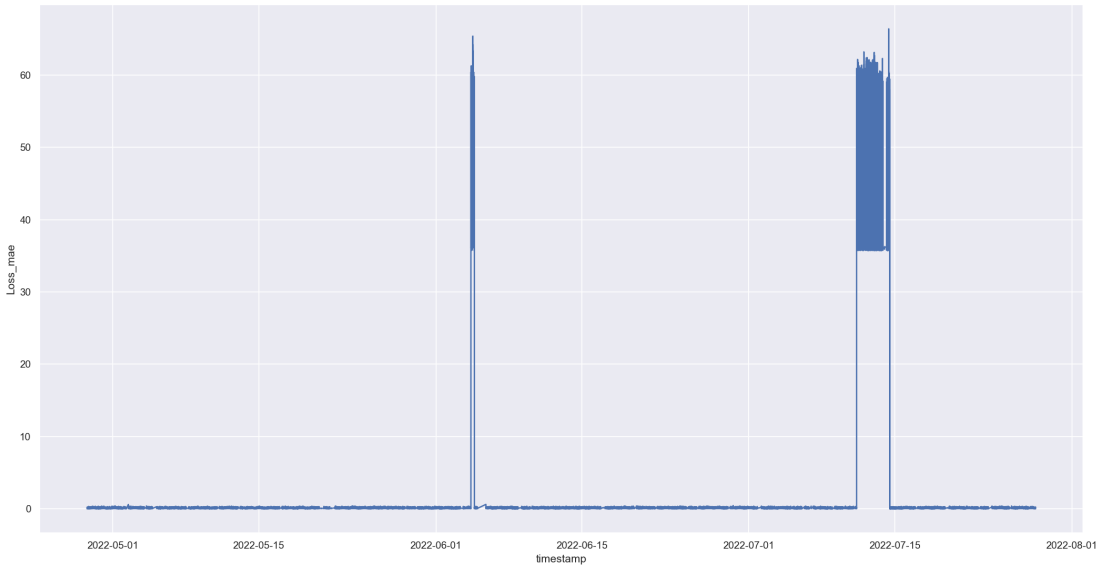

Figure 4.6: Reconstruction Error MetroPTV2

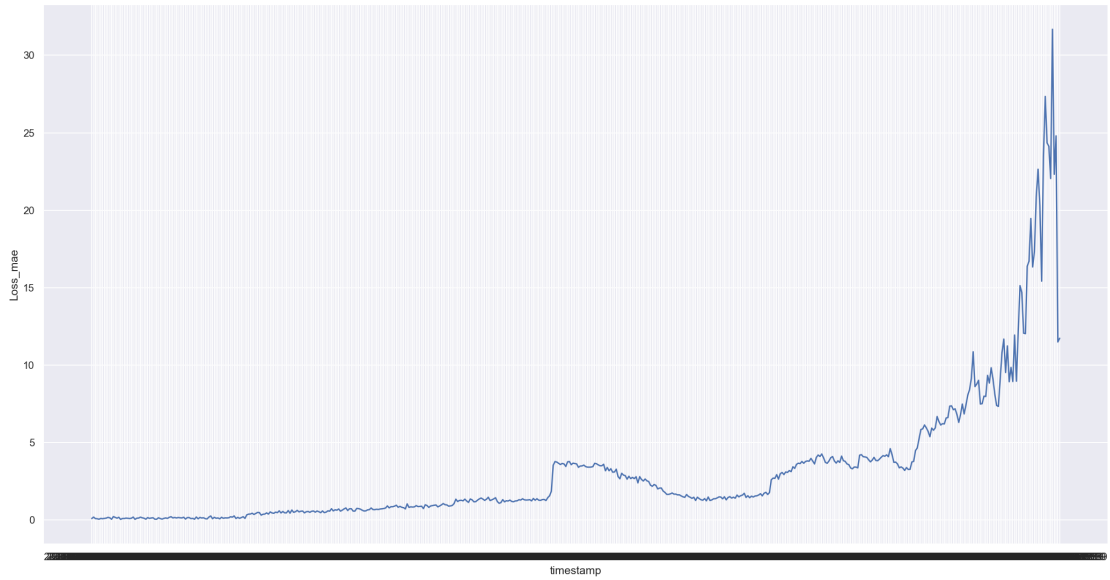

Figure 4.7: Loss function MetroPT v2

many false positives and negatives. F1 Score, which combines precision and recall, indicates that the model is ineffective at detecting anomalies. When employing the second approach, the model presents a precision score of approximately 0.015, while its recall increases to around 0.333, resulting in an F1 score of approximately 0.029. This second approach reinforces the ineffectiveness of the model to detect anomalies.

In evaluating the model trained on the second version of the MetroPT dataset using the first approach, our model demonstrates an approximate precision value of 0.8889. This demonstrates a great ability to reduce false positive predictions, with most errors occurring in the two-hour window preceding and following anomalies. Regarding recall, our model receives a flawless score of 1, demonstrating its remarkable efficiency in identifying all anomalies within the dataset. In addition, the f1-score achieves an exceptional value of 0.941, demonstrating the model's amazing overall performance in anomaly detection. However, in the context of the second approach, which is the one recommended by the dataset, our LSTM Autoencoder excels, receiving a flawless score of 1 across all performance metrics. This extraordinary result can be attributed to the model's ability to predict anomalies two hours in advance and continue illustrating them two hours after their occurrence. In contrast to the first approach, which may classify these predictions as false positives, the second approach recognises them as true positives, highlighting the model's proficiency in anomaly detection.

4.1.4 Discussion

This section examined the results of our LSTM Autoencoders' anomaly detection capabilities. The initial model, trained on the first version of the dataset, failed at detecting the anomalies despite successfully identifying the outliers discussed in the Data Analysis section. This is evident

from the reconstruction error performed without outliers, which failed to highlight anomalies defined by the dataset. Consequently, the evaluation metrics reflect the model's poor performance, yielding unsatisfactory results. The negative results reported could be related to data consistency issues in the dataset. In contrast, our third model, developed for the NASA dataset, reveals a clear machinery deterioration pattern over time, corresponding to the dataset's overall trend. Although the reconstruction error suggests promising potential for anomaly detection, the absence of precise timestamps for the failure events presents a significant obstacle to determining the model's efficacy.

The second LSTM autoencoder, trained using the second version of the MetroPT dataset, stands out with exceptional performance in anomaly detection. This is confirmed by the evaluation metrics, where it attains a perfect score of 1 across all metrics in the second approach. Notably, the model correctly identifies every anomaly in the dataset and predicts anomalies with a lead time of two hours. This flawless performance reaffirms the effectiveness of LSTM Autoencoders as a strategy for predictive maintenance. Based on these results, we can determine that LSTM Autoencoders are an effective strategy for predictive maintenance. Nonetheless, it is essential to emphasise the need for a high-quality dataset.

Based on these findings, we can confidently assert that employing an unsupervised method, such as an LSTM Autoencoder, makes it possible to detect anomalies effectively, offering an exciting opportunity for predictive maintenance applications.

4.2 Can the predictions of a black box model be explained ?

In order to address our second research question regarding the feasibility of explaining the predictions of a Black Box model, such as an LSTM Autoencoder, we will examine the outcomes of the SHAP model in explaining anomalies in the second version of the MetroPT and NASA datasets. Notably, the first version of the MetroPT dataset was excluded from our analysis due to our model's inability to detect anomalies within it, making it unsuitable for anomaly explanation. Subsequently, we present the results produced by the AMRules model, focusing on the explanation of anomalies within the second version of the MetroPT and the NASA datasets. Ultimately, we conclude by discussing the different results attained by each model and how these results help us address the research question.

4.2.1 SHAP

To further develop our understanding of our model's predictions, we will employ the SHAP summary plot to comprehend the influence of individual features on the model's output. In this plot, the vertical axis presents the input features, arranged in descending order of importance, with the most impactful ones at the top. Meanwhile, the horizontal axis represents the average absolute SHAP values assigned to each feature, where dots positioned towards the right indicate a greater impact on the model's predictions. The colour of the dots provides additional information, where blue dots represent low feature values and red dots indicate high feature values. Each summary

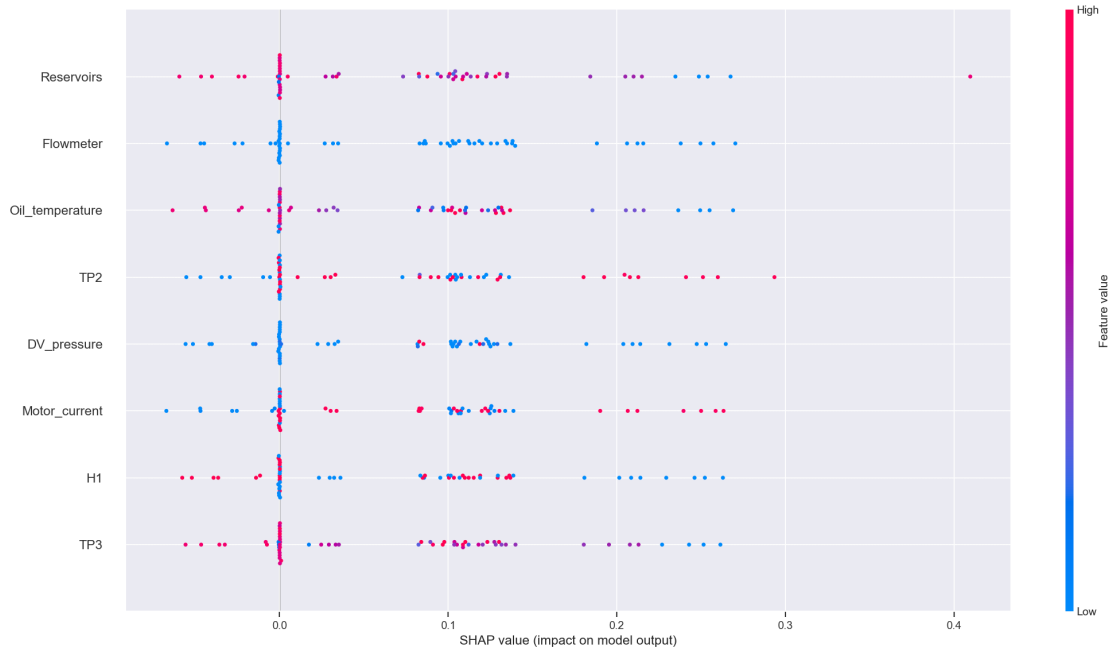

Figure 4.8: SHAP values before the first anomaly

plot will consist of sixty data points, each corresponding to a one-minute time interval recorded by the equipment's sensors in the MetroPT dataset and ten hours in the NASA dataset.

4.2.1.1 First Anomaly from MetroPT 2

To investigate the first anomaly, we will use three graphics. The first graphic, 4.8, displays one minute of sensor data captured thirty minutes before the model detects the anomaly. The second graphic, 4.9, illustrates one minute of data collected during the anomaly occurrence. The third and final graphic, 4.10, illustrates one minute of sensor data collected thirty minutes after the model stopped detecting the anomaly.

This SHAP analysis reveals a distinct pattern. Before and after the anomaly occurrence, individual features' influence on prediction errors is notably reduced, with none exceeding a SHAP value of 0.5. This suggests that during these time intervals, the model's forecasts are relatively stable and not significantly influenced by any one factor. Nonetheless, a significant change occurs during the anomaly period, where all the features' average absolute SHAP values increase significantly, with some reaching close to 500. Additionally, the feature DV_pressure, identified as the most influential feature during this anomaly minute based on SHAP analysis, serves as an example. Before the anomaly, as shown by the majority of blue dots in graphic 4.8, this feature exhibited low values. Nevertheless, as the anomaly develops, the feature DV_pressure experiences a significant increase in values. Even after the anomaly period has ended, its values remain significantly elevated, although their influence on the error diminishes compared to the maximal influence observed during the anomaly.

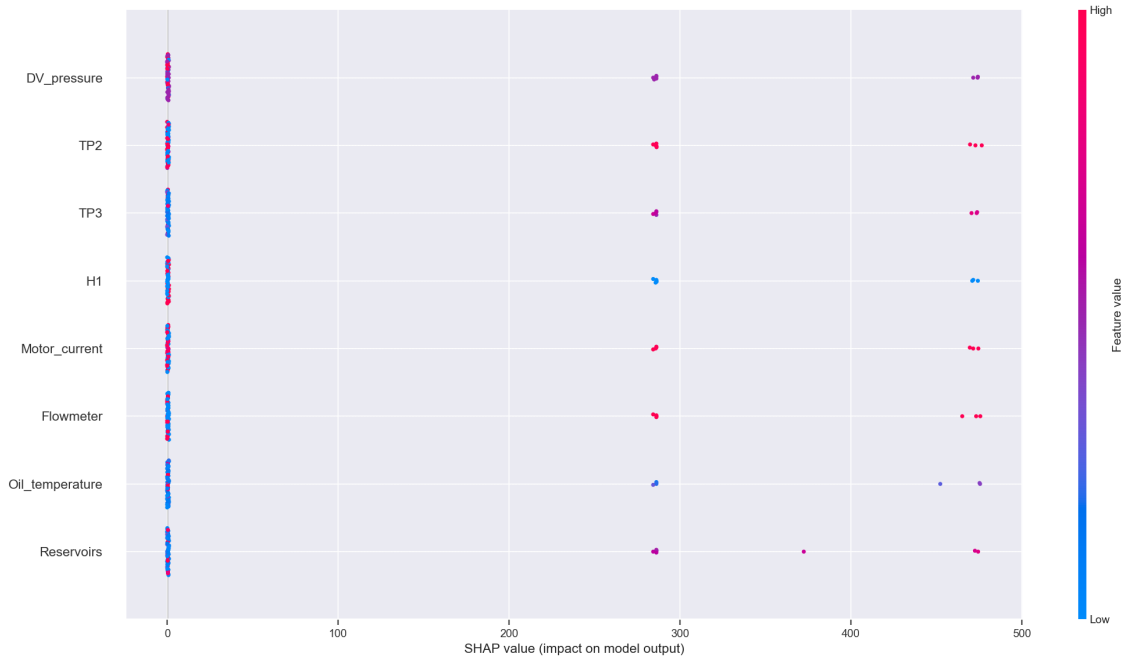

Figure 4.9: SHAP values during the first anomaly

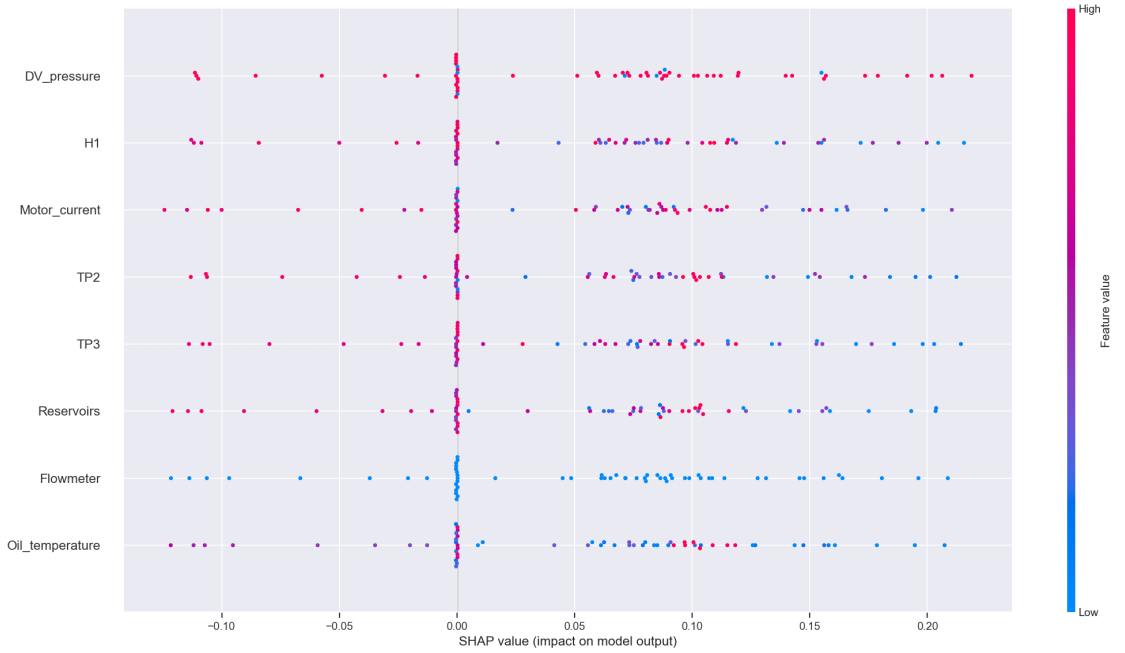

Figure 4.10: SHAP values after the first anomaly

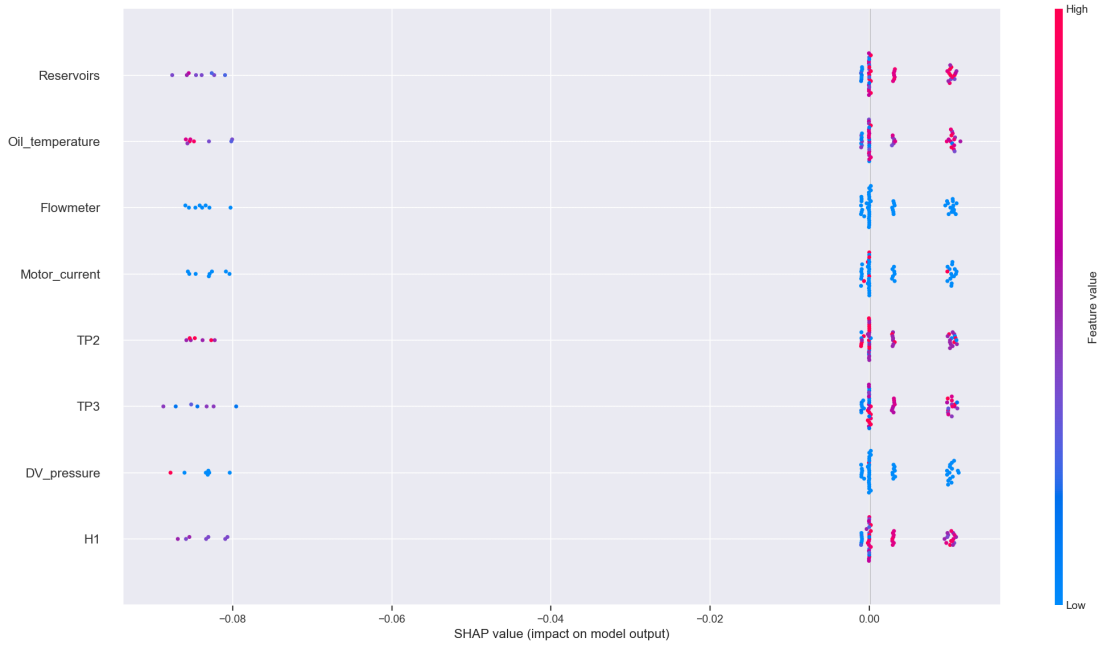

Figure 4.11: SHAP values before the second anomaly

4.2.1.2 Second Anomaly from MetroPT 2

In order to investigate the second anomaly, we will analyse three graphics. The first graph, 4.11, shows one minute of sensor data captured thirty minutes before the model detected the anomaly. The second graphic, 4.12, represents one minute of data collected during the anomaly's occurrence. The third and final graphic 4.13, represents one minute of sensor data recorded thirty minutes after the model finished detecting the anomaly.

This SHAP analysis observes a pattern similar to the first anomaly. The consistently limited influence of individual features on prediction errors suggests that during these time intervals, the model's predictions are relatively stable and not significantly influenced by any feature. However, a change becomes apparent during the anomaly period. During this period, all features' average absolute SHAP values increased significantly, with some values approaching 300. Surprisingly, the most influential feature during this particular anomaly minute is H1, considered the least significant feature before the anomaly. The graphic 4.12 also highlights the importance of low H1 values in determining the model's predictions during the anomaly. Similar patterns are observed in features TP3 and reservoirs, where higher feature values receive a SHAP value of approximately 0. In contrast, lower feature values have a greater impact on the predictions. In addition, it is interesting to note that both the motor current and flowmeter exhibited low values before the anomaly. However, we observe a rise in their values as the anomaly develops. Despite this increase, the SHAP analysis ranks these features lower in terms of their importance during the anomaly.

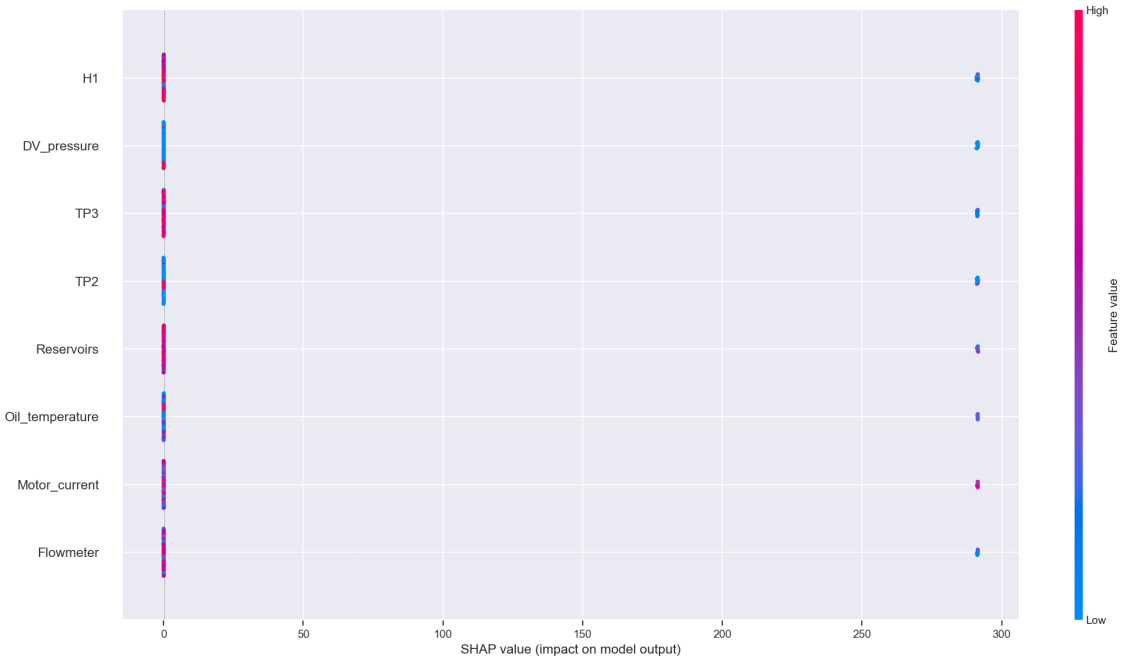

Figure 4.12: SHAP values during the second anomaly

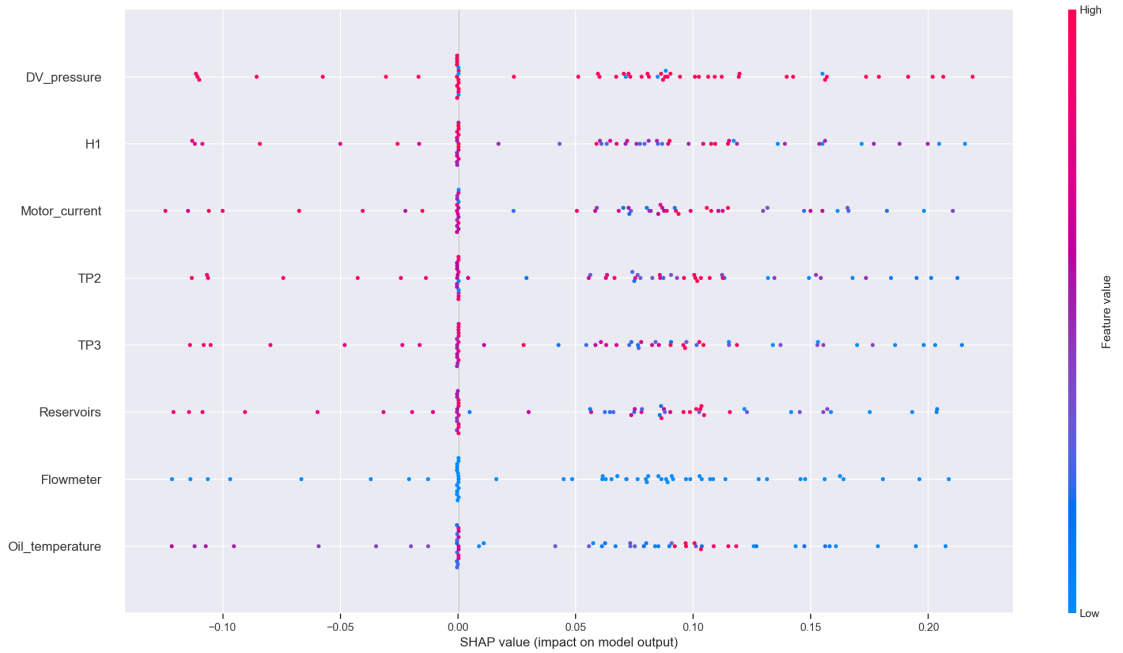

Figure 4.13: SHAP values after the second anomaly

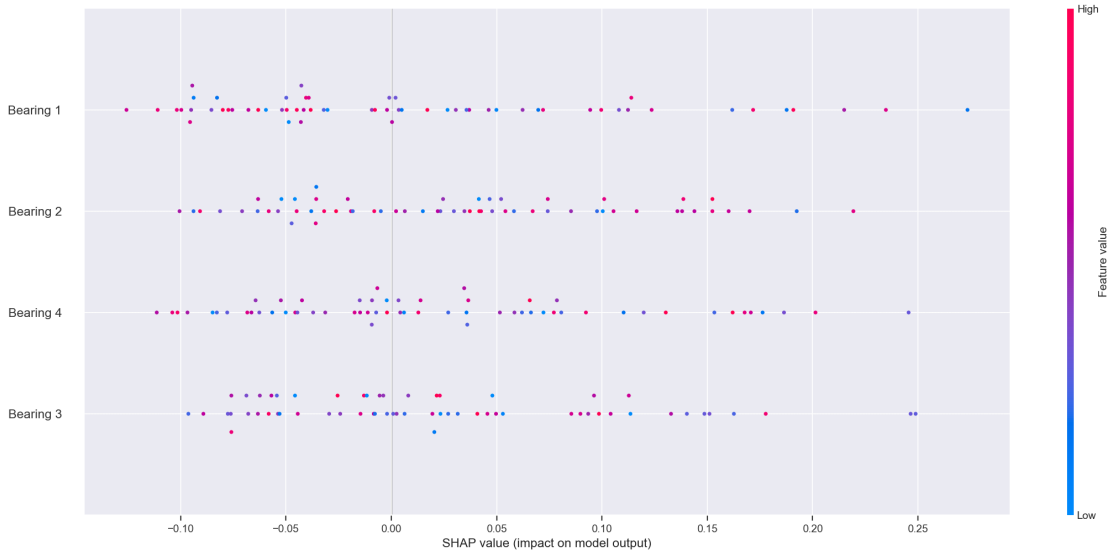

Figure 4.14: SHAP values NASA prestine bearings

4.2.1.3 NASA dataset

We employ two distinct figures to investigate the interpretability of the LSTM autoencoder applied to the NASA dataset. The first, 4.14, represents a 10-hour period when the bearing sensors are in their pristine state, whereas the second, 4.15, illustrates a 10-hour period when the bearing sensors are in their degraded state. In this SHAP analysis, when the bearing sensors are in pristine condition, bearing 1 emerges as the most influential feature. Notably, the dots on the graph share that its influence varies, with a higher value having less influence and a lower value having the most influence. In addition, during the hours the bearing sensors are significantly degraded, bearing 4 emerges as the most significant feature and has the dot with the highest SHAP value. Notably, all three features, bearing 4, 3, and 2, exhibit their greatest impact when their values are at the lower end of the range.

4.2.2 AMRules

In this section, we utilise the capabilities of AMRules to enhance our understanding of the model's operating patterns through rule-based explanations. It is essential to mention that this particular explainability model was only used to explain anomalies. Therefore, we will not provide rule-based explanations for the sensors' normal state. The algorithm operates on a one-second dimension, classifying it with a rule-based explanation learned during the AMRules model's training. In order to facilitate meaningful comparisons during our discussion, we intend to explain the same minute anomaly analysed in the SHAP analysis by examining the rule for each second.

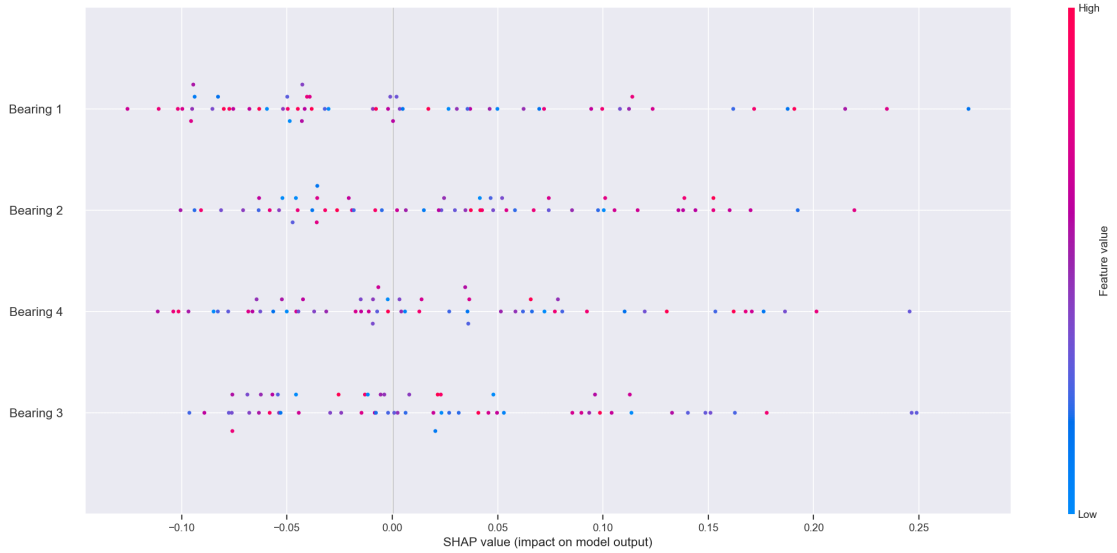

Figure 4.15: SHAP values NASA degraded bearings

4.2.2.1 First Anomaly from MetroPT 2

AMRules employed three rules to explain the one-minute interval within the first anomaly. In the first 26 seconds, the algorithm assigned the first rule, defined by the conditions:

$$TP3 \leq 8.9000 \quad \text{and} \quad Flowmeter \leq 19.9000 \tag{4.1}$$

The figures 4.16 and 4.17 display the sensor signals of the features TP3 and Flowmeter, spanning from one day prior to the occurrence of the anomaly to one day after. It is visible that the TP3 values exhibit a decreasing trend during the anomaly, which explains the first condition of the rule. However, the second condition refers to the flowmeter being less than or equal to 19.9. This condition may give the illusion that the feature is decreasing, whereas an examination of the graphic reveals that it increases during the anomaly. This is because AMRules does not explain the anomaly but focuses on the reconstruction error. Therefore, this condition clarifies why the reconstruction error does not reach higher levels. During this time interval, the reconstruction error has been calculated to be around 36, and it was seen in the previous section that the error could go up to 60.

The model presents a second rule, between the 26th and 38th seconds, with the conditions:

$$Flowmeter \leq 19.2000 \quad \text{and} \quad Motor\_current > 0.0000 \tag{4.2}$$

The first condition of this rule, relating to the Flowmeter feature, is extremely similar to a previously examined rule. The second condition, regarding the motor current being greater than zero, is relevant because in the previous second, the value was approximately -0.00245, whereas in the second 26 it increases to 4.815. The AMRule assigns greater weight to this condition than the previous condition regarding TP3, even though their values are approximately the same.

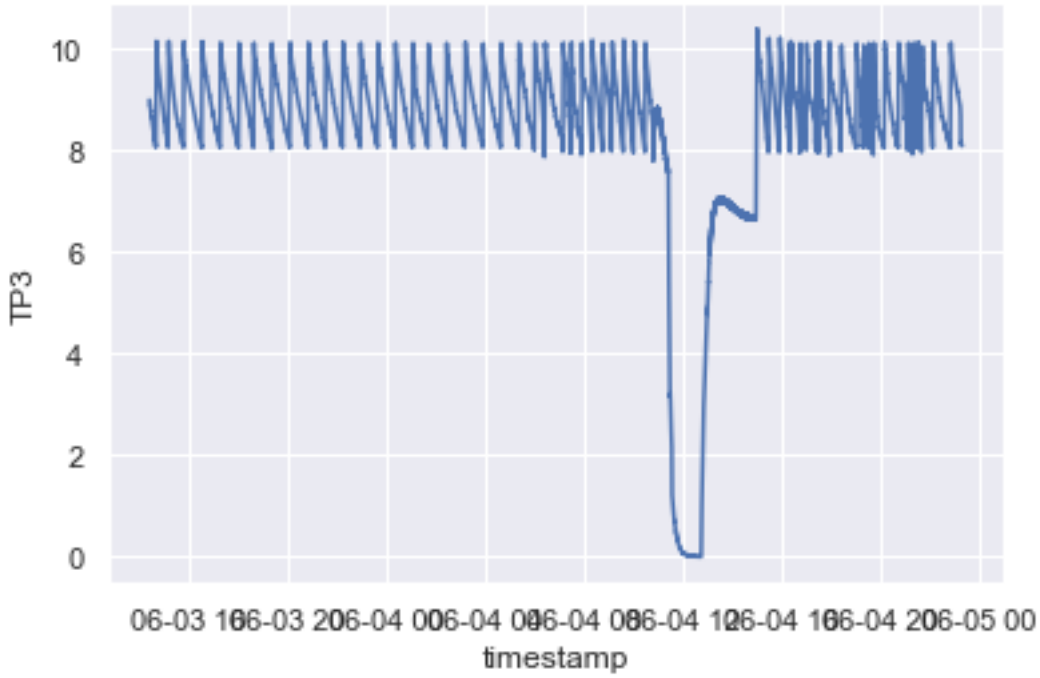

Figure 4.16: TP3 in anomaly 1

The third and final rule, between the 38th second and the end of the minute, is defined by the conditions:

$$Flowmeter > 30.9000 \quad \text{and} \quad Flowmeter \leq 31.3000 \tag{4.3}$$

This rule explains that the flowmeter increased to approximately 31, a significant increase compared to the previous seconds. Additionally, we can observe that the flowmeter is mentioned in all the presented rules, demonstrating this feature's importance in this anomaly.

4.2.2.2 Second Anomaly from MetroPT 2

AMRules utilised two rules to explain the one-minute interval within the second anomaly. The model allocated the rule for the first 4 seconds, defined by:

$$Oil\_temperature > 68.4000 \tag{4.4}$$

The graphic 4.18, which depicts the sensor signal of the oil temperature from one day prior to the anomaly to one day after, demonstrates that the oil temperature increased during the anomaly. Because an oil leak caused the anomaly, the rule-based explanation must underline the relevance of the particular feature. Between the second 4 and the end of the minute, the rule designated by the model is:

$$Flowmeter \leq 26.3000 \tag{4.5}$$

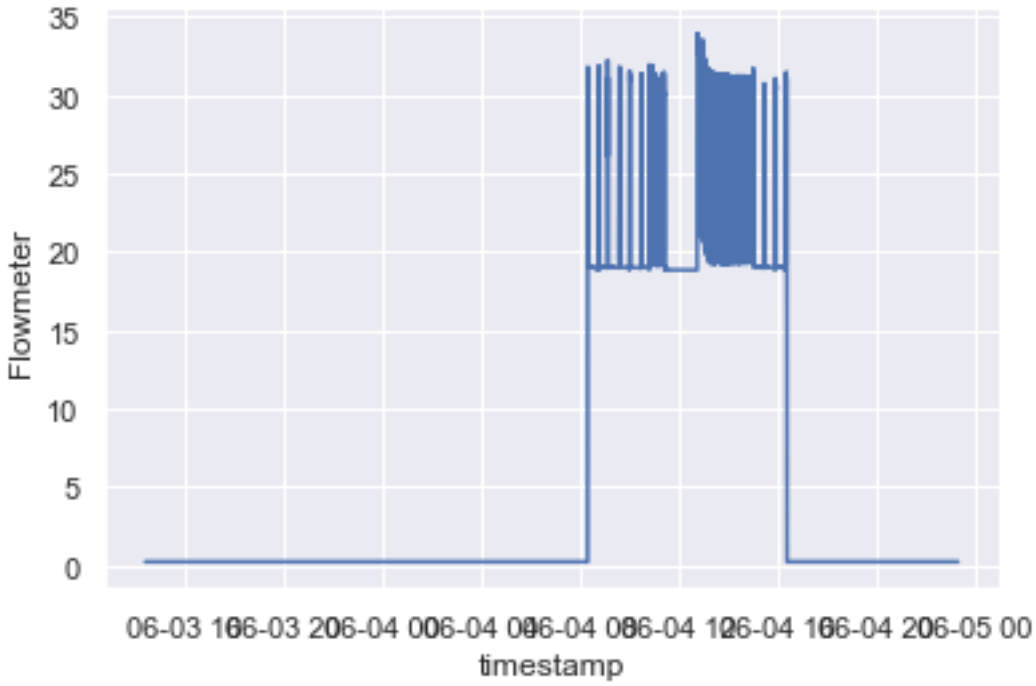

Figure 4.17: Flowmeter in anomaly 1

The graphic 4.19, which represents the sensor signal of the flowmeter from one day prior to the occurrence of the anomaly to one day after, clearly shows an abrupt and notable signal increase, exceeding the high value of 30. Notably, despite this obvious increase, the rule emphasises the importance of the value being below 26.3. We observed a similar behaviour in the first anomaly, in which the rule might give the illusion that the feature is decreasing, revealing its propensity to explain the reconstruction error rather than identifying the anomaly itself.

4.2.2.3 NASA dataset

The initial plan for explaining the predictions of anomaly detection within the NASA dataset was to utilise the 10-hour timeframe previously examined in our SHAP analysis. However, we encountered a problem when attempting to generate explanations using the AMRules model, as it failed to produce any rules. Therefore, we utilise the complete set of test data in order to generate explanations. However, the AMRules algorithm only produced a single rule, with the condition:

$$Bearing1 > 0.1 \tag{4.6}$$

Given the progressive increase in all signals over time, this specific rule explains the reconstruction error. However, this rule, while effective, is generic. Specifically, when we consider the Bearing 1 feature, it attains significantly higher values. Additional rules could have been developed to justify the elevated reconstruction error values comprehensively.

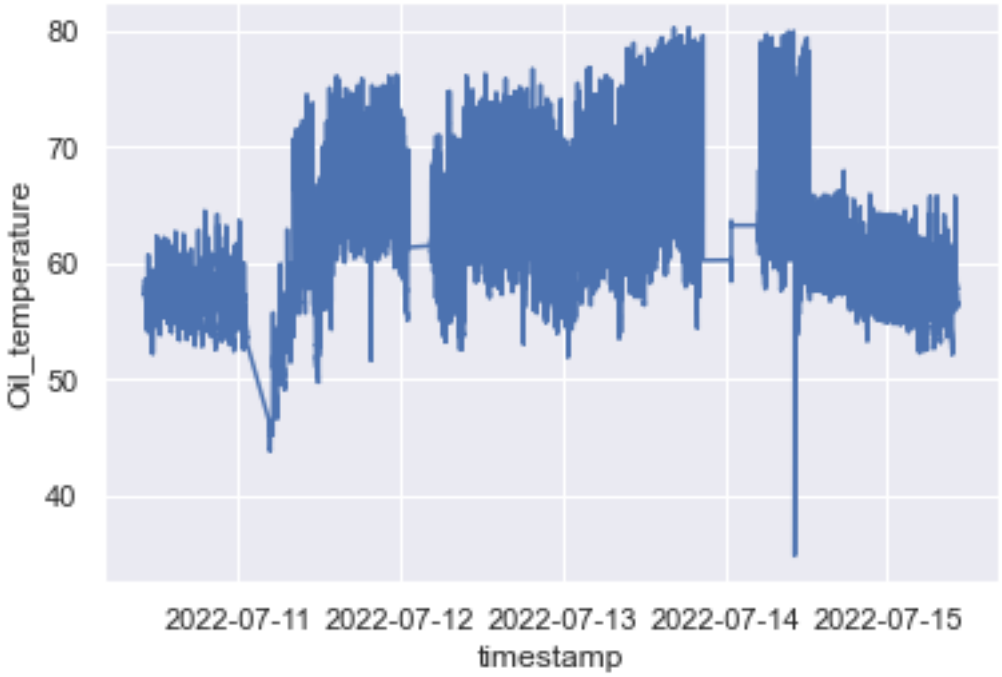

Figure 4.18: Oil Temperature in anomaly 2

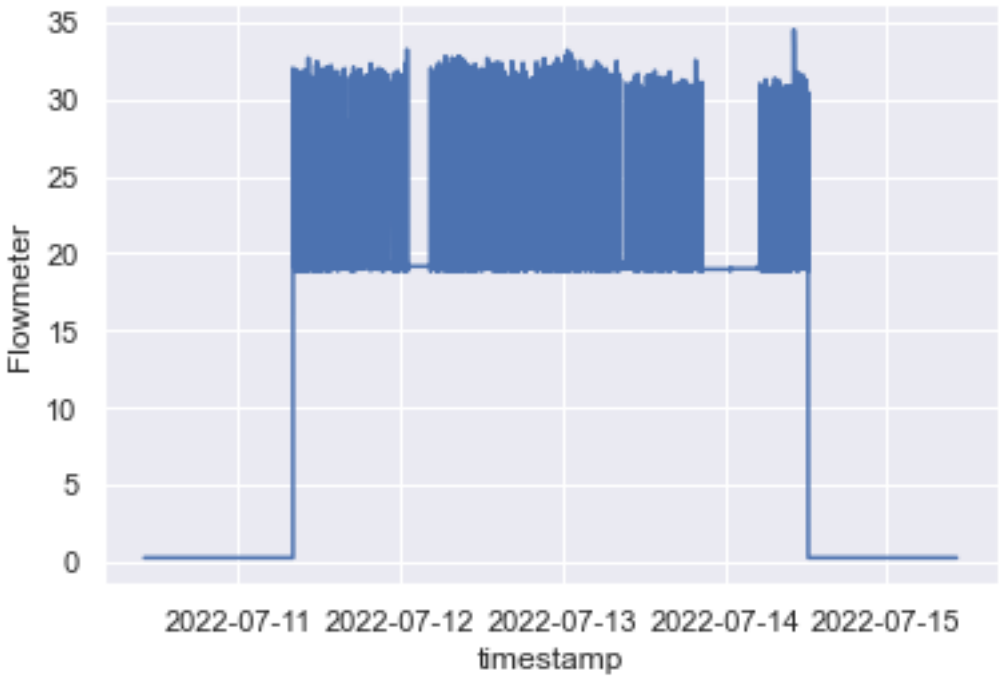

Figure 4.19: Flowmeter in anomaly 2

4.2.3 Discussion

In this section, we analysed the explanations provided by two prominent explainability machine learning models: SHAP and AMRules. Our investigation began with an examination of the SHAP model's generated insights. This model provides a distinctive perspective, enabling us to evaluate the impact of individual features by ranking feature importance in addition to the average absolute SHAP value metric. The SHAP model's graphical representation demonstrates the degree to which each feature influences a single prediction, with the color of each data point indicating whether high or low feature values exert this influence. However, it is essential to note that when working with complex datasets such as MetroPT, where all features carry a great deal of weight, it is difficult to pinpoint the precise causes of anomalies, which is a limitation of the SHAP model.

In contrast, the AMRules model employs a different approach by offering rule-based explanations that are easily interpretable. While these rules excel at finding potential anomaly causes, we have identified a limitation. In particular, the rules emphasise the reconstruction error rather than explaining the underlying cause of the anomaly. Consequently, they may not explain what the cause of the anomaly is but rather why the reconstruction error is not even higher, which can sometimes mislead the audience. In addition, the AMRules model can also be employed as a global explanation technique, providing valuable insights into the features' importance by observing the rules that target higher reconstruction errors.

Both models have different advantages, but their applicability depends on the explanation's audience. If the audience has strong analytic capabilities, the SHAP model can provide more valuable information about model behaviour and the impact of individual features on a specific instance. A straightforward rule-based explanation, such as that provided by AMRules, is more valuable for audiences less versed in data analysis due to its simplicity of comprehension.

Based on these findings, our research demonstrates that it is possible to provide explanations for the predictions of a black box model, in this case, an LSTM Autoencoder, without compromising model performance, which was the primary objective of this investigation.

4.3 Limitations

In this study, we have a few limitations. Our main limitation was insufficient processing power, as we needed more computational capacity for training the SHAP model. This limitation significantly affects the quality and comprehensiveness of the explanations for the anomalies. One further limitation is related to the insufficiency of domain knowledge to evaluate the level to which the provided explanations effectively explained the anomaly.

Chapter 5

Conclusions and Future Work

This concluding chapter highlights the main conclusion derived from our research findings and offers potential directions for future research work.

In conclusion, this dissertation demonstrates successful work in exploring the capabilities of LSTM autoencoders in the context of predictive maintenance. The obtained findings provide robust evidence for the viability of developing an efficient predictive maintenance tool utilising this specific architecture. Another significant finding from our study highlights the importance of data quality for successfully implementing LSTM autoencoders in predictive maintenance. Furthermore, our study highlights the importance of including interpretability techniques such as SHAP and AMRules in development. These techniques provide informative explanations for the predictions made by black box models while maintaining their effectiveness. This represents significant progress toward improving the accessibility and comprehensibility of complex models for a broad spectrum of individuals. Thus, this dissertation makes a valuable contribution to the expanding field of research on predictive maintenance and the efficient utilisation of deep learning models in practical contexts.

While significant progress has been made in understanding and implementing predictive maintenance models emphasising explainability, future research should investigate the explainability of other complex black box models, such as genetic algorithms and deep reinforcement learning models. Other possible future works can investigate new explanation methods and assess the precision of the explanations concerning domain knowledge.

References

[1] S Agatonovic-Kustrin and Rosemary Beresford. Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research. *Journal of pharmaceutical and biomedical analysis*, 22(5):717–727, 2000.

[2] Ezilda Almeida, Carlos Ferreira, and Joao Gama. Adaptive model rules from data streams. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part I 13*, pages 480–492. Springer, 2013.

[3] Shideh Shams Amiri, Sam Mottahedi, Earl Rusty Lee, and Simi Hoque. Peeking inside the black-box: Explainable machine learning applied to household transportation energy consumption. *Computers, Environment and Urban Systems*, 88:101647, 2021.

[4] Xanthi Bampoula, Georgios Siaterlis, Nikolaos Nikolakis, and Kosmas Alexopoulos. A deep learning model for predictive maintenance in cyber-physical production systems using lstm autoencoders. *Sensors*, 21(3):972, 2021.

[5] Alejandro Barredo-Arrieta, Ibai Laña, and Javier Del Ser. What lies beneath: A note on the explainability of black-box machine learning models for road traffic forecasting. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2232–2237. IEEE, 2019.

[6] Marcus Bengtsson and Gunnar Lundström. On the importance of combining "the new" with "the old"–one important prerequisite for maintenance in industry 4.0. *Procedia manufacturing*, 25:118–125, 2018.

[7] Sailendu Biswal and GR Sabareesh. Design and development of a wind turbine test rig for condition monitoring studies. In *2015 international conference on industrial instrumentation and control (icic)*, pages 891–896. IEEE, 2015.

[8] Torsten P Bohlin. *Practical grey-box process identification: theory and applications*. Springer Science & Business Media, 2006.

[9] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[10] Thyago P Carvalho, Fabrízzio AAMN Soares, Roberto Vita, Roberto da P Francisco, João P Basto, and Symone GS Alcalá. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137:106024, 2019.

[11] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[12] Tiago Dos Santos, Fernando JTE Ferreira, João Moura Pires, and Carlos Damásio. Stator winding short-circuit fault diagnosis in induction motors using random forest. In *2017 IEEE International Electric Machines and Drives Conference (IEMDC)*, pages 1–8. IEEE, 2017.

[13] Tai Dou, Benjamin Clasie, Nicolas Depauw, Tim Shen, Robert Brett, Hsiao-Ming Lu, Jacob B Flanz, and Kyung-Wook Jee. A deep lstm autoencoder-based framework for predictive maintenance of a proton radiotherapy delivery system. *Artificial Intelligence in Medicine*, 132:102387, 2022.

[14] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[15] Hardik A Gohel, Himanshu Upadhyay, Leonel Lagos, Kevin Cooper, and Andrew Sanzetenea. Predictive maintenance architecture development for nuclear infrastructure using machine learning. *Nuclear Engineering and Technology*, 52(7):1436–1442, 2020.

[16] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*, pages 986–996. Springer, 2003.

[17] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[18] Timo Huuhtanen and Alexander Jung. Predictive maintenance of photovoltaic panels via deep learning. In *2018 IEEE Data Science Workshop (DSW)*, pages 66–70. IEEE, 2018.

[19] Sheikh Rabiul Islam, William Eberle, Sid Bundy, and Sheikh Khaled Ghafoor. Infusing domain knowledge in ai-based" black box" models for better explainability with application in bankruptcy prediction. *arXiv preprint arXiv:1905.11474*, 2019.

[20] Ameeth Kanawaday and Aditya Sane. Machine learning for predictive maintenance of industrial machines using iot sensor data. In *2017 8th IEEE international conference on software engineering and service science (ICSESS)*, pages 87–90. IEEE, 2017.

[21] 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.

[22] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[23] Octavio Loyola-Gonzalez. Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view. *IEEE Access*, 7:154096–154113, 2019.

[24] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[25] Freddie Mac. Single family loan-level dataset. *Freddie Mac. Available online: http://www. freddiemac. com/research/datasets/sf_loanlevel_dataset. page (accessed on 3 February 2019)*, 2019.

[26] S Matzka. Ai4i 2020 predictive maintenance dataset. *UCI Machine Learning Repository*, 2020.

[27] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[28] Christoph Molnar. A guide for making black box models explainable. *URL: https://christophm. github. io/interpretable-ml-book*, 2018.

[29] Todd G Nick and Kathleen M Campbell. Logistic regression. *Topics in biostatistics*, pages 273–301, 2007.

[30] Cecilia Panigutti, Alan Perotti, and Dino Pedreschi. Doctor xai: an ontology-based approach to black-box sequential data classification explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 629–639, 2020.

[31] Marina Paolanti, Luca Romeo, Andrea Felicetti, Adriano Mancini, Emanuele Frontoni, and Jelena Loncarski. Machine learning approach for predictive maintenance in industry 4.0. In *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, pages 1–6. IEEE, 2018.

[32] T Praveenkumar, M Saimurugan, P Krishnakumar, and KI Ramachandran. Fault diagnosis of automobile gearbox based on machine learning techniques. *Procedia Engineering*, 97: 2092–2098, 2014.

[33] Arun Rai. Explainable ai: From black box to glass box. *Journal of the Academy of Marketing Science*, 48(1):137–141, 2020.

[34] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[35] Rita P Ribeiro, Saulo Martiello Mastelini, Narjes Davari, Ehsan Aminian, Bruno Veloso, and João Gama. Online anomaly explanation: A case study on predictive maintenance. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 383–399. Springer, 2022.

[36] Subham Sahoo, Huai Wang, and Frede Blaabjerg. On the explainability of black box data-driven controllers for power electronic converters. In *2021 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 1366–1372. IEEE, 2021.

[37] Lloyd S Shapley. A value for n-person games. *Classics in game theory*, 69, 1997.

[38] Neeraj Sharma and Mala Kalra. Predictive maintenance for commercial vehicles tyres using machine learning. In *2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–6. IEEE, 2022.

[39] Parvathaneni Naga Srinivasu, N Sandhya, Rutvij H Jhaveri, and Roshani Raut. From black-box to explainable ai in healthcare: existing tools and case studies. *Mobile Information Systems*, 2022, 2022.

[40] 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(3):812–820, 2014.

[41] Gero Szepannek and Karsten Lübke. Explaining artificial intelligence with care. *KI-Künstliche Intelligenz*, pages 1–10, 2022.

[42] Vagan Terziyan and Oleksandra Vitko. Explainable ai for industry 4.0: Semantic representation of deep learning models. *Procedia Computer Science*, 200:216–226, 2022.

[43] Andrea Torcianti and Stephan Matzka. Explainable artificial intelligence for predictive maintenance applications using a local surrogate model. In *2021 4th International Conference on Artificial Intelligence for Industries (AI4I)*, pages 86–88. IEEE, 2021.

[44] Jasper van der Waa, Elisabeth Nieuwburg, Anita Cremers, and Mark Neerincx. Evaluating xai: A comparison of rule-based and example-based explanations. *Artificial Intelligence*, 291:103404, 2021.

[45] Bruno Veloso, Rita P Ribeiro, João Gama, and Pedro Mota Pereira. The metropt dataset for predictive maintenance. *Scientific Data*, 9(1):764, 2022.

[46] Simon Vollert, Martin Atzmueller, and Andreas Theissler. Interpretable machine learning: A brief survey from the predictive maintenance perspective. In *2021 26th IEEE international conference on emerging technologies and factory automation (ETFA)*, pages 01–08. IEEE, 2021.

[47] Yuyi Zhang, Feiran Xu, Jingying Zou, Ovanes L Petrosian, and Kirill V Krinkin. Xai evaluation: Evaluating black-box model explanations for prediction. In *2021 II International Conference on Neural Networks and Neurotechnologies (NeuroNT)*, pages 13–16. IEEE, 2021.