National College of Ireland

Optimizing Metro Vehicle Maintenance: A Deep Learning Framework for Failure Prediction of Critical Components

MSc Research Project
Data Analytics

Tulio Begena Araujo
Student ID: 22133721

School of Computing
National College of Ireland

Supervisor: Musfira Jilani

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Tulio Begena Araujo
Student ID:	22133721
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Musfira Jilani
Submission Due Date:	14/12/2023
Project Title:	Optimizing Metro Vehicle Maintenance: A Deep Learning Framework for Failure Prediction of Critical Components
Word Count:	6233
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Tulio Begena Araujo
Date:	14th December 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Optimizing Metro Vehicle Maintenance: A Deep Learning Framework for Failure Prediction of Critical Components

Tulio Begena Araujo
22133721

Abstract

Predictive Maintenance (PdM) is the state-of-the-art strategy for maintenance management. It is able to provide cost reduction in various sectors, including manufacturers, power plants, and transportation systems. This research proposes a new PdM framework that could be used by railway operations to enhance their vehicles maintenance strategy. This was done adapting the Cross Industry Standard Process for Data Mining (CRISP-DM) for the specificities of this project. The framework comprises four tiers: Data Acquisition, Data Transfering, Data Processing, and Decision Making. A Deep Learning model coupling Long Short-Term Memory with Autoencoder was developed and tested with a recent published dataset built with real data from sensors. The metrics reached in this study were 65% of Recall, 28% of Precision, and 40% F1 Score. These results, together with the method used to get them, mean that the proposed framework can be a viable strategy for implementing PdM.

1 Introduction

Maintenance constitutes a huge portion of costs for businesses relying on mechanical equipment, such as manufacturers, power plants, and transporters. Traditional maintenance approaches Are known as Run-to-Failure (R2F) and Preventive Maintenance (PvM), and both have well-known drawbacks (Carvalho et al.; 2019; Çınar et al.; 2020). R2F, or corrective maintenance, means that an equipment will be repaired (or replaced) after it stops working. This can lead to equipment failures, unplanned downtime, or even accidents. PvM, or scheduled maintenance, is a technique which maintenance are made periodically, aiming to prevent equipment failures. This technique result in unnecessary maintenance, increasing operational costs. In this scenario, a new method was proposed and have increasing attention from researchers in recent years: the Predictive Maintenance (PdM). PdM means predicting equipment failures. It offers a more efficient alternative by predicting when equipment maintenance is truly necessary, minimizing failures and downtime (Schwendemann et al.; 2021). This proactive approach relies on dynamic scheduling of maintenance plans according to faults detected. Other advantages that can advent from failures predictions are decrease in accidents, decrease of pollution, and decreasing labor costs (Zhang et al.; 2019). Implementing PdM can lead to cost savings of up to 30% in factories and improve operational safety (Cakir et al.; 2021).

The forth revolution in industry that humanity is creating, denominated Industry 4.0, relies on big datasets, obtained through sensors installed in a variety of equipment (Serradilla et al.; 2022), making possible the tasks involved in a PdM strategy. Machine learning methods development provided a way to utilize all this generated data, from sensors monitoring vibration, sound, oil-analysis, among others (Frankó et al.; 2022). While data collection has become easier with advancements in information technology, challenges remain, including network bandwidth, data quality, and integration system issues (Traini et al.; 2019). Establishing correlations between sensor data and equipment failures can be challenging, and data access is often restricted due to privacy concerns.

Research on PdM has gained traction in recent years, driven by advancements in machine learning (ML) and deep learning (DL) algorithms and sensors used (Namuduri et al.; 2020). In the realm of public transport, faults during regular operations can result in significant damages, especially when they cause trip cancellations (Veloso, Gama, Ribeiro and Pereira; 2022). Early detection of such faults is crucial to avoid disruptions and maintain trust in transportation services. Several publications have addressed data-driven PdM and the application of these techniques in the railway industry (Veloso, Gama, Ribeiro and Pereira; 2022). Key areas of research focus on failure prediction, remaining useful life (RUL) estimation, and root cause analyses (RCA).

The implementation of PdM is specific for each case. This makes it challenging and leaving a vast space to improve in this sense, until industries can solely trust in such predictions. This research aims to contribute to the field by answering the question:

How imminent component failures can be accurately predicted in air pressure systems of trains to avoid unnecessary downtimes of the vehicles, reduce costs from operations, and prevent trip cancellations, utilizing sensor data coupled with deep learning algorithms?

Investigating the application of deep learning algorithms to identify imminent component failures in trains is worth exploring because the implementation of proactive maintenance strategies can significantly reduce operational problems, minimize unexpected stops, and enhance the overall reliability and safety of train systems. In this context, the initial investment on sensors installation, internal network, data preprocessing, and database design, is high and determines the quality of the data. However, still many tasks involving the ML and DL application need to be completed to achieve the desired results, such as classification, clustering, regression, dimensional reduction, and system design (Frankó et al.; 2022). The development of the best model is crucial for the PdM implementation, since the gain of this maintenance method is dependent of the capability of the model to correctly predict component failures (Zhang et al.; 2019).

The proposed research question is feasible since there are existing deep learning algorithms that have shown promising results in various domains, including PdM. These algorithms can be applied to analyze the vast amount of sensor data collected from trains, enabling the detection of patterns and correlations that indicate impending component failures. An unsupervised learning approach is even more attractive at this point, since it saves people work of labeling data. On top of that, this research propose an anomaly detection strategy to forecast failures. Additionally, tools such as Python-based libraries (e.g., TensorFlow, Keras) and deep learning frameworks provide accessible resources for testing and implementing the algorithms. At this rate, Autoencoders (AE) algorithms, coupled with Long Short-Term Memory (LSTM), are the option analysed. The success of the research can be evaluated based on the algorithm's performance in terms of F1

score, precision, and recall. This ensures that the research outcomes can be objectively measured and compared to previous published works.

The remainder of this paper is structured as follows: in Section 2 a literature review is constructed, first introducing the need for new PdM approaches, then with subsections more focused on PdM frameworks, and on the algorithms used: AE and LSTM. Section 3 comprehend the steps taken to conclude this proposed project. In this section the dataset used for the experiments is presented in details and it is explained how the data was handled and with which tools. Section 4 describes the proposed framework. The outputs, the proposed model architecture, and the code written are examined closely in Section 5. The most relevant results are displayed and evaluated in 6. The last section (Section 7), closes this paper with a discussion and a summary of all its content and what can be done further to enrich this research.

2 Related Work

In the last years the terms industry 4.0 and industrial internet of things gained relevance, referring to the increasing availability of data that can describe in real time, among other parameters, the state of mechanical equipment. The correct use of these data can lead to increase in revenue, extension of lifetime of systems, and improvement of security (Frankó et al.; 2022). These advantages reinforce the importance of more researches in this area and add to the recent attention given by industries on artificial intelligence and machine learning techniques (Serradilla et al.; 2022).

Overall, the cited studies highlight the need for innovative PdM approaches. This is due to the fact that it is very difficult to apply generic frameworks in this field with such a wide range of possibilities. In the next subsection several approaches will be explained based on previous studies. Following to next subsections, the justifications of the chosen algorithms used in this research are given.

2.1 PdM Frameworks and Algorithms

PdM can be applied to a wide range of business. Due to this characteristic it can be tricky to adapt the framework developed for a specific case to new problems. Hence, many researches propose new frameworks and test them with datasets from the field of interest (when available).

(Zhuang et al.; 2023) worked on a general PdM framework based on Bayesian DL. For the case study, the famous C-MAPSS dataset (made available by NASA Ames Prognostics Center of Excellence) was used. The model applied was evaluated by Root mean square (RMSE), accuracy, and score, a metric that penalize late predictions more than early predictions. Despite not having the entire life cycle for PdM application, e.g., data collection and storage, the proposed framework includes simulations of operational constraints, something often neglected when using simulated datasets. Also, other algorithms could be used to compare with the final solution results. The same dataset was used by (Nguyen and Medjaher; 2019), that proposed a PdM framework including the decision-making process with the results obtained from the model prediction. The decision process used the probability of a system failure occur in a certain time window. Hence, the probabilistic confusion matrix was used to evaluate the results from predictions. Only one algorithm was tested, a LSTM classifier. (Liu et al.; 2021) proposed a likewise framework with a LSTM algorithm, but using generative adversarial network

(GAN). In this case, the proposed PdM was tested with data collected from a manufacturing system and different models were compared to verify the robustness of the proposed system: Convolutional Neural Network-Long-Short Term Memory, Global Average Pooling Convolutional Neural Network and the Wasserstein Generative Adversarial Networks. All the models achieved more than 90% accuracy, with LSTM-GAN generating the best result. These results therefore need to be interpreted with caution, whereas other evaluation metrics could be applied, like confusion matrix that was applied only for the LSTM-GAN algorithm. Nevertheless, the proposed framework was capable of predicting future failures successfully.

Yujie Wang et al(Wang et al.; 2022) compared different RNN algorithms detecting anomalies in some train systems. The GRU, LSTM, and their new algorithm performed much better than traditional RNN, the last of the three being a little better.Silvestrin, Hoogendoorn, and Koole (Silvestrin et al.; 2019) applied LSTM and Temporal Convolutional Network (TCN) to predict failures based on the Hydraulic System Sensor dataset. The DL algorithms were compared with ML classic algorithms, like Random Forest, Decision Tree, and k-nearest neighbors, regarding the classification error (one minus accuracy). This study concluded that DL algorithms can perform poorly if not enough data is available. In this sense, TCN needed less data than LSTM to reach a good performance. (Wu et al.; 2020) used LSTM for a PdM. This work shows that the motivations to implement a PdM strategy are vast and can be, e.g., on the environmental consequences that can be derived from equipment failures. The LSTM algorithm and SVM method were applied to a dataset obtained from accelerometers installed on four bearings connected to a shaft. SVM showed similar performance to LSTM in prediction of health state of the equipment. On the other hand, while predicting more critical states, the superiority of LSTM was verified and illustrated by confusion matrices.

Most frameworks are focused on the model performance. While this can be determinant for the success of a new maintenance strategy, two other components of a complete framework should not be neglected: the preprocessing of the acquired data and how the predictions can bring real value for the decision-making process.

Coupling AE with CNN was the strategy of (Gatta et al.; 2022) to predict the health situation of oil wells into eight classes. The CNN-AE was used to extract features from the raw data and input them into different ML algorithms, namely RF, KNN, Gaussian Naive Bayes, and Quadratic discriminant analysis. The contribution of this work was the hybrid approach, using DL and ML to obtain better results. In the work of (Davari et al.; 2021), two types of AE were applied: sparse AE (SAE) and variational AE (VAE). The aim was to implement PdM to predict failures in the air pressure unit of trains, based on anomalies detection, with two hours in advance, at least. This method discard the need of external feedback, which can facilitate the implementation of such strategy. The results were evaluated by recall, precision, and F1 score. In this case, SAE algorithm performed better than VAE in all three metrics.

One of the advantages of the AE model is that it is capable to detect anomalies without be trained with them, i.e., perform anomaly detection by unsupervised learning. The study conducted by (Abdelli et al.; 2022) compared many DL algorithms. The data used was derived from experiments monitoring the aging of semiconductor tunable lasers. The proposed model combines GRU and attention mechanisms to predict the degradation of the lasers, uses AE to detect anomalies and to extract statistical features related to them, and, finally, group this information into a fully connected layer which its output is the RUL. This approach was compared with MLP, CNN, RNN, and LSTM DL models,

besides RF and SVR. The proposed method outperformed all the others, resulting in the lowest RMSE. It is valid to note that the computational time of the proposed method was higher than all the others.

Two researches made use of LSTM coupled with Autoencoders (AE)(Bouabdallaoui et al.; 2021; Bampoula et al.; 2021). The former study aimed to propose a generic framework for building installations. Using AE, this was made without the need of labeled data, i.e., unsupervised learning. The conducted experiments, however, was important for the definition of a threshold, representing how big a detected anomaly should be treated as a predicted failure. This framework contemplates stages since data collection until model improvement from experimental feedbacks and exhibited capacity to predict real failures two days before they happen. However, only two failures were confirmed during the study, an issue for many studies that use real world experiments. The second study (Bampoula et al.; 2021) used labeled data, despite the use of AE. This innovative approach consisted in separating the labeled data into smaller datasets, one for each label (representing high, medium, and low level of equipment health status), and building a LSTM-AE model from each dataset. The drawback is the requirement expert knowledge to be feasible. To evaluate a new input, it should be tested against the three models and the classification result would come from the model with the best metric. The case study used data collected from a rolling mill machine used for metal bars production. The authors put good effort improving the preprocessing of the raw real data, which usually shows incongruousness. The predictions obtained from the LSTM-AE model could decrease the preventive stoppages of the equipment by 22.2%.

It is notable that the studies solving a classification problem in this subsection evaluated the models performance with at least precision, recall, and F1 score, showing how common these metrics are in this field of study. This is not a surprise owing to the fact that equipment are expected to fail as few times as possible, producing unbalanced datasets. Notwithstanding, it is paramount to reinforce the fact that late predictions in real world can generate much higher damage (e.g. cost, safety, and efficiency) than early (or extra) predictions. In this sense, a metric that considers this can be used, as in (Zhuang et al.; 2023).

It is clear the high capability of DL models to predict failures using data available from the state of equipment. In addition, however, the quality and the quantity of available data are of major importance for the final outcome, and it is still a challenge. In light of the above, the data used in (Davari et al.; 2021) is of exceptional importance for benchmark in this field, whereas it contains large amount of observations and a relatively number of failures reported.

2.2 Long Short-Term Memory

LSTM is a type of RNN, designed to improve long-time dependencies solving the problems of vanishing and exploding gradients. While neurons in a traditional RNN use memory from one time step before as input with the new data to process sequences, LSTM networks use a system of gates (Figure 2), which permit it to learn and keep most important features for long-term (Hochreiter and Schmidhuber; 1997). In this recent study (Martínez-Llop et al.; 2023), it was showed how the use of sequential algorithms perform better than time independent in a sensor based prediction.

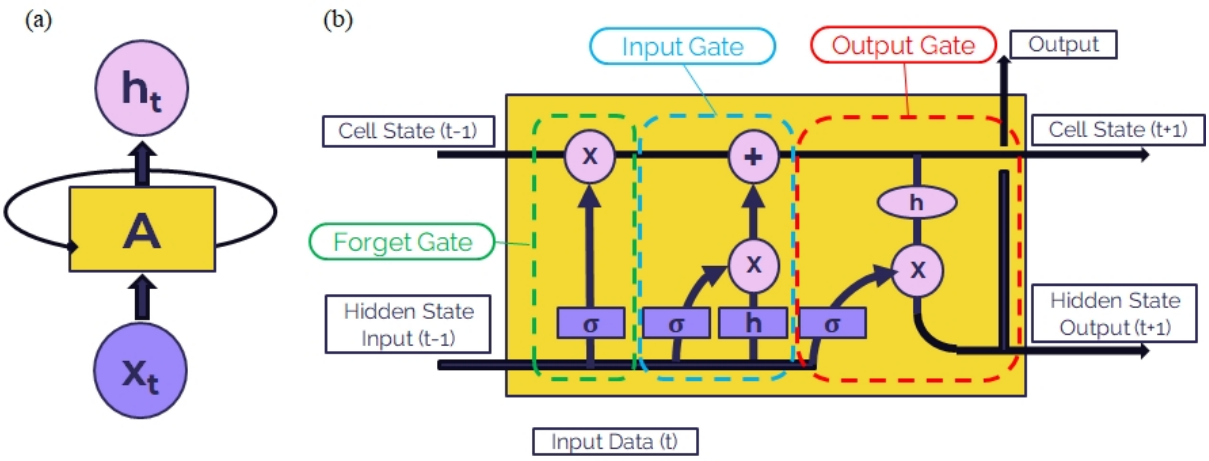

Figure 1: (a) RNN and (b) LSTM cells.

2.3 Autoencoder

Autoencoders are designed to reconstruct the input data, as close as possible, using only the most important features. To accomplish this, this algorithms are built with two main parts: the encoder and the decoder. Each part can contain a variable number of layers. Between them, itpically there is a bottleneck, that forces the network to learn only from the essential features (Namuduri et al.; 2020). The Figure **??** is a generic representation of an AE.

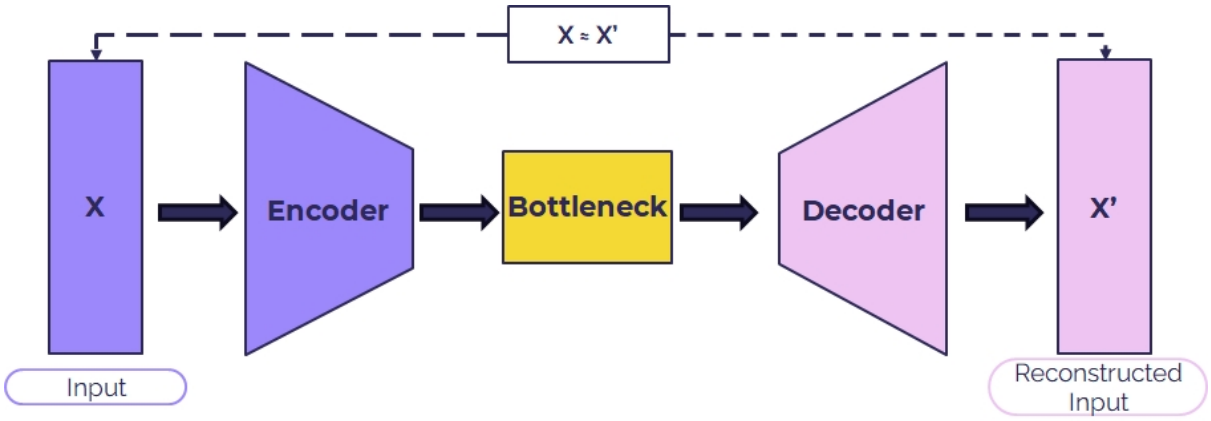

Figure 2: Generic AE representation.

This design forcing the network to learn the features is a form to automate the feature engineer process. One disadvantage is that the explainability of the model is compromised (the infamous "black box"). On the bright side, it permits the approach to be data-driven in all its phases.

2.4 Key Insights

This literature review provides precious insights into the field of PdM. A variety of frameworks were presented. More often then not, the process of PdM comprises data source,

data storage, preprocessing, features extraction, model training, forecast. For each domain, or case, the frameworks need to be adapted, creating demand for more studies to fill this gaps.

Many proposed frameworks tested in real world experiments showed potential to decrease the downtime of equipment, vehicles, or systems. Some used simulated constraints, elevating the complexity of the approaches while getting closer to real scenarios. Some of the frameworks incorporated an valuable topic: based by equations and predictions, data-driven decision making processes were proposed.

Moreover, various DL models, highlighting LSTM and AE, have been explored by researchers. These models are capable to predict future failures in many systems with remarkable precision, meaning that it is feasible the implementation of a framework to avoid unnecessary downtimes of metro vehicles coupling these techniques. Since the prediction power of applied models are a prominent part of the PdM process, it is a good practice to couple some of these techniques aiming to optimize the outputs, as indicated by many researches.

Metrics like recall, precision, and F1 score have been widely used to assess the effectiveness of the models in detecting anomalies and predicting impending failures. They are important due the fact that the datasets are imbalanced and the goal is to difficulty is to decrease the number of False Positives in the predictions.

The main goal of the research is to propose a PdM strategy for train components. Taken together, this review suggest that precise predictions of component failures can lead to significant reductions in downtime, operational costs, and trip cancellations, ultimately enhancing the efficiency and safety of train operations. This can be done by implementing a PdM strategy, with a framework designed with attention to data characteristics and an optimal model that combine LSTM and AE approaches to score high regarding the discussed metrics.

3 Methodology

The present research aims to process sensor data into deep learning algorithms to establish a framework that can anticipate and detect potential failures in train components. This framework is to provide decision-making support through a data-driven approach. The steps applied in this study, to produce knowledge from data, were adapted from the Cross Industry Standard Process for Data Mining (CRISP-DM)(Schröer et al.; 2021) and are described in the following subsections.

3.1 Business and Data Understanding

Data created for PdM is expected to be high unbalanced. It is also expected that the results from anomaly detection generate a high number of False Positives, and this issue needs to be addressed further. The less False Positives generated by the model, less purposeless and costly stops should be made by train operators.

The dataset used is denominated MetroPT-3 Dataset (Davari and Gama; 2023). It was donated to UC Irvine Machine Learning Repository by its authors (Davari et al.; 2021). This dataset consists of multivariate time series data, dating between February and August 2020, obtained from analogue and digital sensors installed on the compressor of a train in Porto, Portugal. A total of 1516948 observations from 15 sensors is available. Each sensor is referenced as one feature (7 from analogue sensors and 8 from digital

sensors), such as oil temperature, pressure, and electrical signals. Detailed information about the how this data was collected and the air pressure system is available in the author's original publication (Veloso, Ribeiro, Gama and Pereira; 2022).

The data is unlabeled, making it suitable to anomaly detection by an unsupervised learning approach. To better evaluate the results, 21 reported failures are described in (Davari et al.; 2021), with start and end times, as can be seen in Table 1.

Table 1: Reported failures.

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

3.2 Data Preparation and Modelling

The Data Preparation and Modelling were grouped in one topic to permit a clearer explanation of the procedures of this research. This choice was made for the reason that the training of the model was made using sliding windows. As is illustrated in Figure 3, a window 8 days long is splitted into train and test in the proportion 7:1. This means a week of data is used to train a model to detect anomalies one day ahead. It is assumed in this research that detecting anomaly in this time window makes it possible to forecast a failure in the subsequent day, in the next 2 days, or in the next 3 days. This assumptions were tested in the implementation. The example with assumption of forecasting only in the subsequent day is showed as 'Forecast' in the Figure 3.

The windows were set moving one day ahead from the previous one. In this way, one prediction is produced for each day from the first Forecast day.

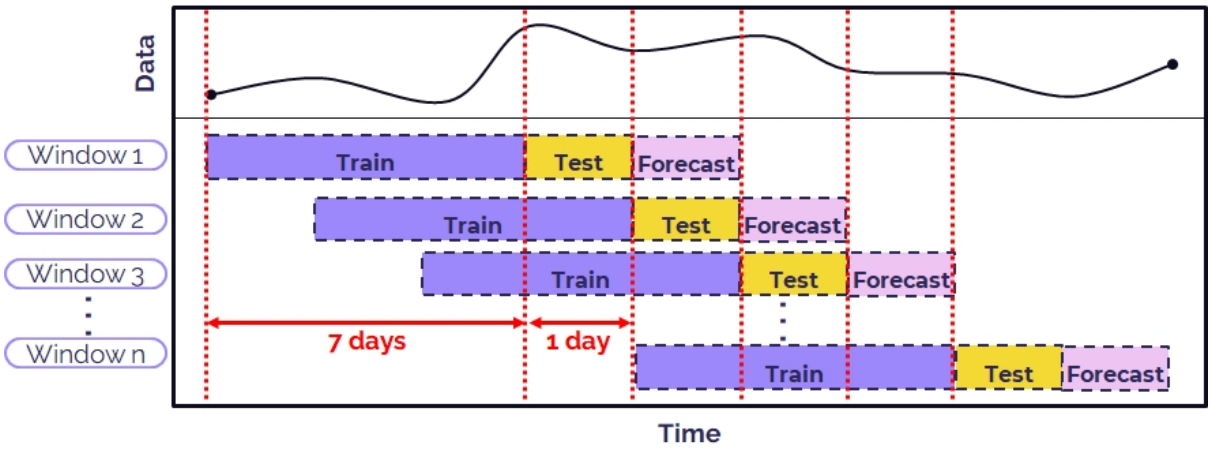

Figure 3: Representation of sliding windows schema.

3.2.1 Preprocessing

There was no missing values in the entire dataset. The distribution of intervals between observations was analysed, since there is variation between their values. The dataset was regularized and resampled to frequencies of observations equal to 20 and 60 minutes. The value for the observations were calculated as the mean from the original observations included in this range. This generated new dataframes with null values, because there were intervals between original observations larger than 20 minutes and some larger than 60 minutes.

That being the case, longer intervals with null values were deleted, because it was assumed they refer to times when the train was not in a journey. While the sequence of the data is of paramount importance for modelling, this is not the case for the exact time of the observations, e.g., if a failure occurred on Wednesday or Friday. Isolated observations with null values had these values replaced by the next non-null values.

3.2.2 Training

A few models coupling LSTM and AE were tested in this research, varying the settings (or hyperparameters). For the first window of a test, the data was normalized using the function MinMaxScaler (scikit-learn library) from -1 to 1. To complete this, scaler.fit_transform and scaler.transform were applied in the train and test sets, respectively. The arrays obtained were reshaped to three dimensions using Numpy library to fit as input in the LSTM-AE network. Subsequently, the model was trained using the input data as target. Prediction was made for the test set having itself as the target. Then, reverse scaling was used in the output and computed the mean absolute error (MAE) for the train and test. Finally, the MAEs were compared. The MAE of the training was used as the threshold, i.e., it is the limit of error that separates an anomaly from a normal event. If the test returned a MAE higher than the training, this test window was classified as an anomaly.

For the next window, the start of training test is moved one day ahead, an so on until the last window available (as already showed in Figure 3.

3.3 Evaluation and Deployment

The models were evaluated by the metrics recall, precision, and F1 score, which are calculated as shown in Equations 1, 2, and 3, respectively.

$$Recall = TP/(TP + FN) \tag{1}$$

$$Precision = TP/(TP + FP) \tag{2}$$

$$F1\ Score = 2 * Precision * Recall/(Precision + Recall) \tag{3}$$

It is necessary, to calculate these metrics, to define True Positive (TP), False Positive (FP), and False Negative (FN). This is illustrated in Figure 4. If a window detect an anomaly in the test set, forecasting a failure, and in the subsequent day a failure was reported, it is a True Positive. If, in this case, there is no report of failure in the next day, it is a False Positive. If a window do not detect an anomaly and a failure is reported in the next day, it is a False Negative.

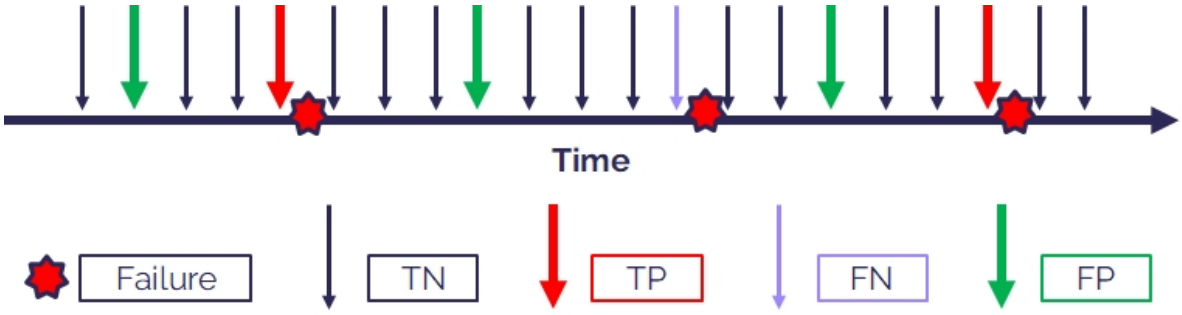

Figure 4: Performance definition.

A proposed framework to drive business value from the knowledge produced is described in the next section, together with the architecture of the best model tested.

4 Design Specification

The proposed framework architecture for a dynamic Predictive Maintenance is shown in Figure 5. The higher level of this framework encompasses the tiers Data Acquisition, Data Transfer, Data Processing, and Decision Making.

4.1 Data Acquisition

This part consists in collect data from analog and digital sensors installed in a critical component of the train, capable of measuring different aspects in real time, with frequency between 1 and 0.2 Hz, i.e. one observation for each 1 to 5 seconds. The definition of the component (or components) to be monitored depends on the necessity of each case, considering the severity of failures caused to the whole system and possible losses that come with non predicted failures. In this study, the component is the air pressure unit of a train. From each sensor a feature is generated and recorded with its timestamp, producing time series

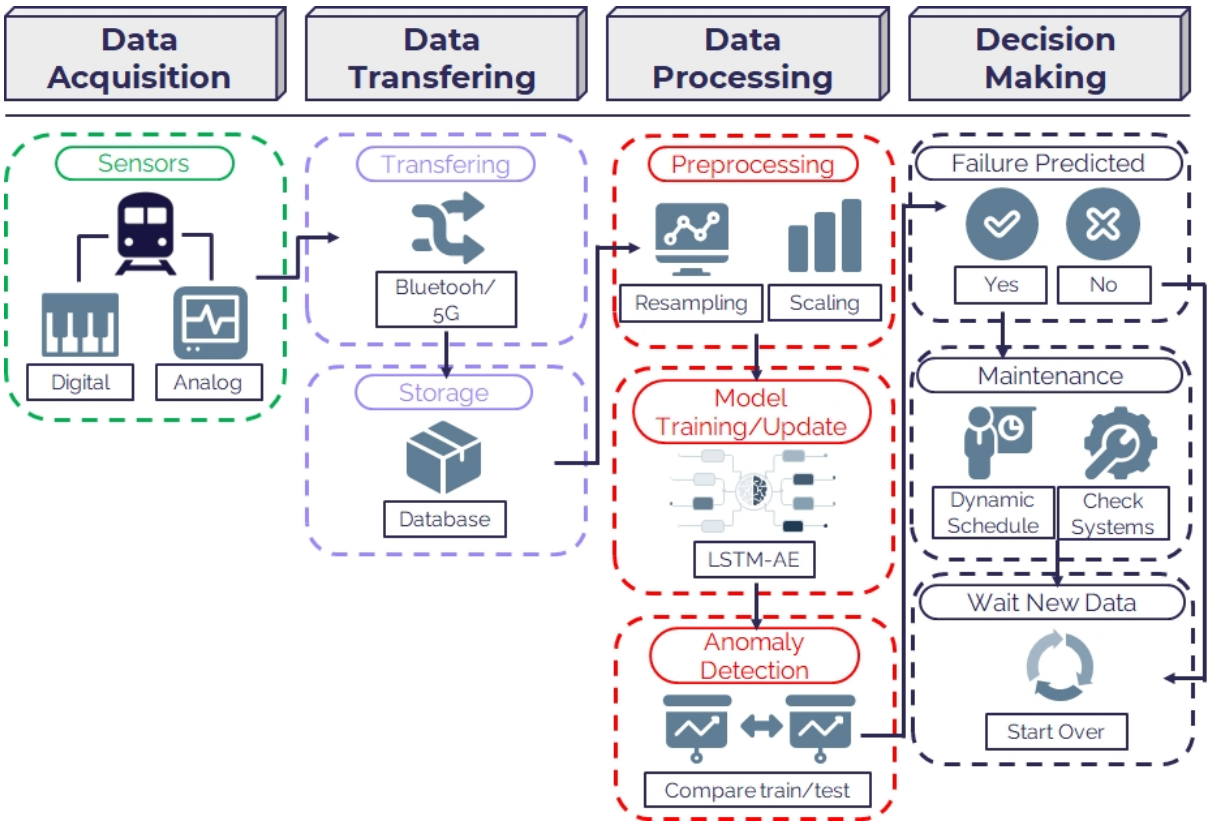

Figure 5: Online Data-Driven Predictive Maintenance Framework.

4.2 Data Transfer

Data gathered from sensors is initially transferred via Bluetooth to a compiler device in the train. From it, the data is transferred every five minutes to the main database, using the 5G network. If this database is local or a cloud environment, it depends on the project implementation.

4.3 Data Processing

The two last steps are the focus of this study. From the data stored in the database, the last 8 days are processed. Firstly, the data is resampled to longer intervals between observations than it is available, e.g. 60 minutes. The resample is important for three reasons. First, to normalize the time interval between observations. Second, to decrease the computation power needed for training the model. Third, to make data compatible with the best model developed, since the time between observations is a hiperparameter that needs to be tuned during model development. The resampled dataset is, then, scaled to be used as input in the model to be trained/updated. The architecture of the model that best fits the studied case is presented in Figure 6. The model is a LSTM-AE network, in which the encoder consists of 2 layers. The first layer contains 4 units and the second layer contains 2 units. The decoder have 2 layers, mirrored from the encoder.

In the start of implementation, the model is trained with data from a period that failures did not occur. This is to build a model that can distinguish regular data from anomaly data. At least a month of regular data is recommended for this process.

Layer (type)	Output Shape	Param #
lstm_112 (LSTM)	(None, 504, 4)	208
lstm_113 (LSTM)	(None, 2)	56
repeat_vector_28 (RepeatVector)	(None, 504, 2)	0
lstm_114 (LSTM)	(None, 504, 2)	40
lstm_115 (LSTM)	(None, 504, 4)	112
time_distributed_28 (TimeDistributed)	(None, 504, 8)	40

Total params: 456 (1.78 KB)
Trainable params: 456 (1.78 KB)
Non-trainable params: 0 (0.00 Byte)

Figure 6: Architecture parameters.

For the LSTM-AE model building/update, the data used correspond from the first to seventh days, named as training set. The MAE resulting of the training is calculated. Next, the data available from the last day is used as input for the model to have its MAE calculated. Both MAE from training set and from test set are used to define the anomaly score.

4.4 Decision Making

The anomaly score is then evaluated. If the anomaly score is negative, it means a failure is predicted for the next 24. 48, or 72 hours. In this case, a maintenance is scheduled to check on the air pressure unit.

Independent of the failure prediction, new data is constantly acquired from the sensors. In the moment that enough new data is available in the database to make a new test set, the Data Processing step is done again, including the update of the model, and another window of prediction is available for the Decision Making process.

5 Implementation

The specification of the computer which the experiments were ran has the following specification: AMD Ryzen 5 2500U with Radeon Vega Mobile Gfx, 2.00 GHz, and 8 GB RAM. A Python 3 kernel in Jupyter Notebook App was used to load the data, starting with the reading of dataset's CSV file through Pandas library, and its conversion to dataframe format.

The proposed framework was tested using the MetroPT-3 dataset. Sliding widows were created to simulate a procedure of acquiring new data and updating the model every day. After resampling the dataset, however, many rows with null values were created. This happened because there were large time gaps between some observations.

The analysis of these gaps showed that the distribution of intervals with null values over the hours of the days occur between 22 pm and 5 am. This is shown in Figure 7. These intervals are likely to be due the train is not in a journey. The small gaps were filled with the next non value from the dataset, while the largest gaps were just deleted.

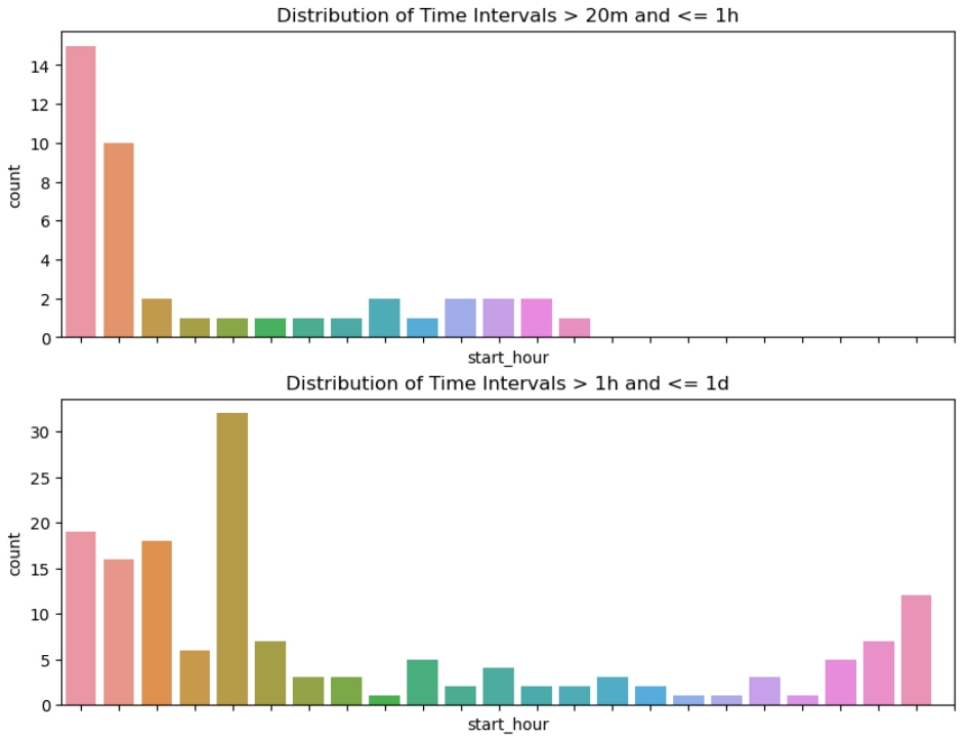

Figure 7: Hourly distribution of gaps in the dataset.

A total of 24 tests were performed for a initial analysis. The variations were the resample time (20 and 60 minutes), LSTM layers in encoder/decoder, and units in them (1 layer with 128 and 4 units, 2 layers with (128, 64) and (4, 2) units each, and 3 layers with (128, 64, 32) and (8, 4, 2) units each). Finally, all these settings were tested using features from the analog and digital sensors separately. The hyperparameters of each test can be seen in Table 2. The metrics resulting from each test are in the same Table.

6 Evaluation

The models tested were evaluated to increase the comprehension about Predictive Maintenance strategy. The variations tested were shown in the previous section.

6.1 Initial models

Overall, the precision of the models were low. This means that the models returned a high number of False Positives. This is a common issue when dealing with failures detection. Also, this led to low F1 Scores. If implemented in a real environment, this would elevate the costs of this maintenance strategy. On the other hand, a few models returned, paired with this False Positives, a high number of True Positives. This favors the reliability in the framework and increase the safety of the operations.

Table 2: Hyperparameters tested in experiment.

Model	Layers	Freq.	Sensors	Units	TP	Recall	Precision	F1
0	1	60	Analog	128	7	35	14	20
1	1	60	Analog	4	6	30	7	11
2	1	60	Digital	128	16	80	10	18
3	1	60	Digital	4	12	60	10	18
4	1	20	Analog	128	na	na	na	na
5	1	20	Analog	4	18	85	11	20
6	1	20	Digital	128	17	80	11	19
7	1	20	Digital	4	15	71	12	21
8	2	60	Analog	(128, 64)	9	45	22	30
9	2	60	Analog	(4, 2)	18	90	16	27
10	2	60	Digital	(128, 64)	na	na	na	na
11	2	60	Digital	(4, 2)	9	45	10	17
12	2	20	Analog	(128, 64)	3	14	15	14
13	2	20	Analog	(4, 2)	na	na	na	na
14	2	20	Digital	(128, 64)	na	na	na	na
15	2	20	Digital	(4, 2)	12	57	14	22
16	3	60	Analog	(128, 64, 32)	na	na	na	na
17	3	60	Analog	(8, 4, 2)	2	10	10	10
18	3	60	Digital	(128, 64, 32)	na	na	na	na
19	3	60	Digital	(8, 4, 2)	12	60	11	19
20	3	20	Analog	(128, 64, 32)	2	9	6	8
21	3	20	Analog	(8, 4, 2)	4	19	13	16
22	3	20	Digital	(128, 64, 32)	na	na	na	na
23	3	20	Digital	(8, 4, 2)	11	52	15	23

6.2 Final models

The models 8, 9, 15, and 23 were analysed further, since presented the best results. The model 8 (named 2260Aup) have 2 LSTM layers in the encoder, with 128 and 64 units, analog features, and dataset resampled to 60 minutes. The model 9 (named 2260Ado) have 2 LSTM layers in the encoder, with 4 and 2 units, analog features, and dataset resampled to 60 minutes. The model 15 (named 22D20do) have 2 LSTM layers in the encoder, with 4 and 2 units, digital features, and dataset resampled to 20 minutes. The model 23 (named 33D20do) have 3 LSTM layers in the encoder, with 8, 4, and 2 units, digital features, and dataset resampled to 20 minutes.

These models were tested changing the scaler from MinMax scaler to Standard scaler, but no improvements were observed. They were also tested in relarion of the window of forecast. The results are in Tables 3, 4, and 5.

Analysing the three tables, an expected result is noticed. The metrics are better for longer the forecast window. In this sense, would be necessary an in deep cost analysis to define which forecast window present better results. Analysing only the metrics, the 3 days forecast window has the best results. Virtually all failures were detected. The models 22D20do and 33D20do are equivalent. This is due the fact that the only difference between them is that the second one has one more layer than the former one. This extra

Table 3: Results from best models with 1 day forecast window.

Model	TP	Recall	Precision	F1 Score
2260Ado	8	40	13	20
22D20do	11	52	15	23
33D20do	11	52	15	23

Table 4: Results from best models with 2 days forecast window.

Model	TP	Recall	Precision	F1 Score
2260Ado	12	50	20	28
22D20do	17	60	23	33
33D20do	17	60	23	33

layer, however, has the same amount of units than the input and output layers.

6.3 Discussion

The models tested were capable to detected the failures before they occur. The chosen model is the model 15, 22D20do, since it is equivalent with model 23, but without the purposeless extra layers. In Figure 8 the test loss is showed over the sliding windows. As the model is updated, the test loss increases. But after 100 windows, it seems it becomes stable.

In Figure 9, the threshold used is showed. The negative values refer to the predicted failures. Negative blue bars are the False Positives, while the red are True Positives. On the other hand, the positive blue bars are True Negatives, while the red ones are the False Positives.

Some False Positives have much longer bars than True Positives. These bars are coincident with the biggest gaps in the dataset. Some other approaches should be tested to deal with these gaps, like using mean or median values.

Overall, the framework works, with a minimal number of failures non detected. The study of (Davari et al.; 2021) showed better metrics. 47%, 90%, and 62% for Recall, Precision, and F1 Score. A 47% Recall, however, means that half of the failures are not predicted, and this can be disastrous for the operations. Actually, the Recall of this study is higher (65%). The metrics are dependent of the definition used for TP, TN, FP, and FN. The issue with FP, that decreased the Precision of this study, needs to be better adressed. Applying some kind of filter in the output of the models could improve these results.

Table 5: Results from best models with 3 days forecast window.

Model	TP	Recall	Precision	F1 Score
2260Ado	19	61	32	42
22D20do	21	65	28	40
33D20do	21	65	28	40

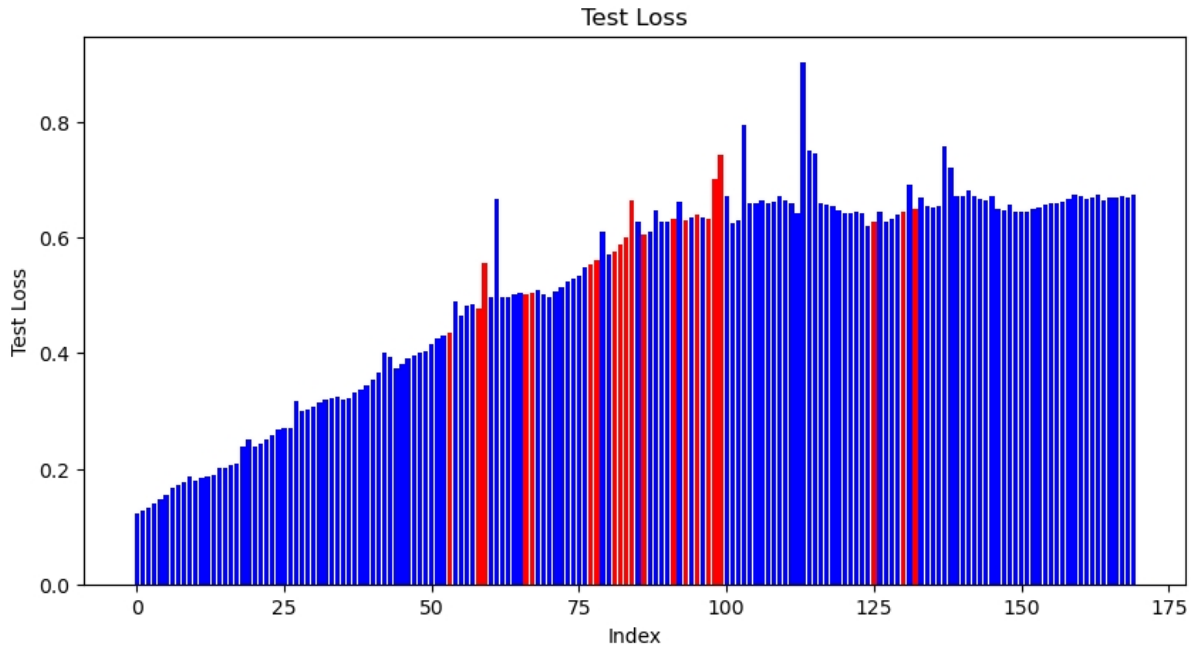

Figure 8: Test loss over windows. In red, the real failures.

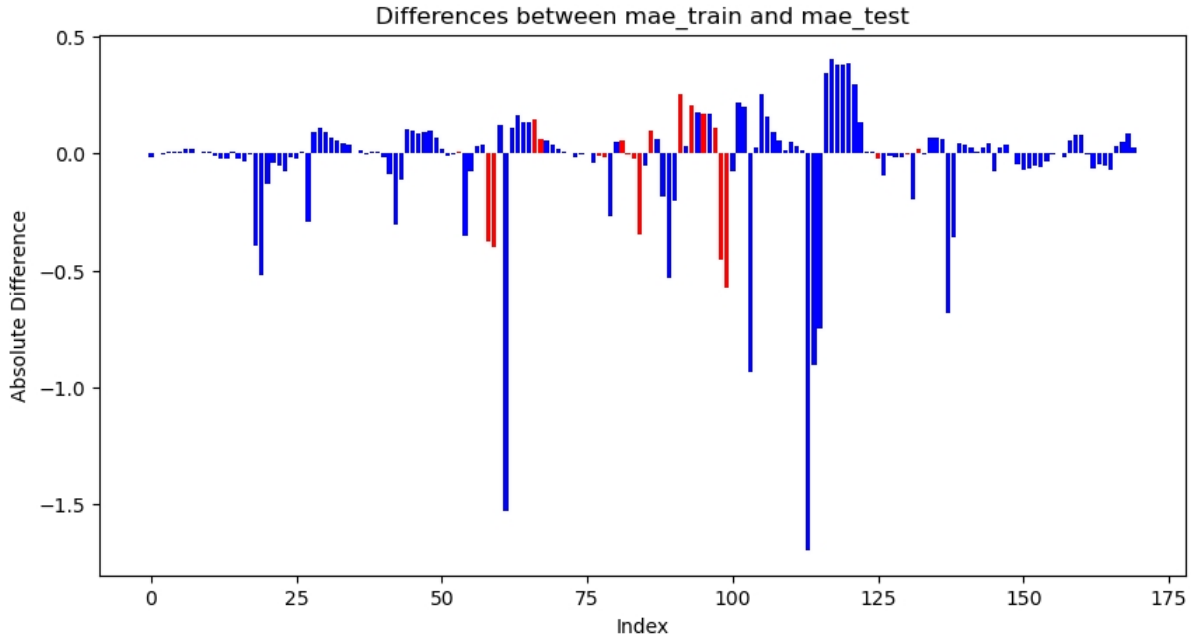

Figure 9: Difference between train MAE and test MAE over the windows. In red, the real failures.

7 Conclusion and Future Work

This research proposed the development of a Predictive Maintenance (PdM) framework for metro vehicle components. This is a important strategy to decrease unnecessary downtimes, operational costs, and to prevent trip cancellations. The methodology outlined a systematic approach to fulfill this objective.

With the test of many models, the best model reached 65%, 28%, and 40% Recall, Precision, and F1 Score. Analysing these metrics coupled with the high True Positives, can be said this research reached its goal. The proposed framework, despite presenting many limitations, is presented as a good option to implement Predictive Maintenance in a train system.

The bigger limitation here comes from the higher number of False Positives generated by the model. This issue should be addressed in future work. One possible solution is testing different types of filters in the output of the model.

Another improvement could come from using a dataset with longer time span. As can be seen by the evolution of test loss, it seems it stabilized. It is possible, due the nature of the equipment used to gather the data, that future updates of the model would be affected by model drift.

References

Abdelli, K., Grießer, H. and Pachnicke, S. (2022). A machine learning-based framework for predictive maintenance of semiconductor laser for optical communication, *Journal of Lightwave Technology* **40**(14): 4698–4708.

Bampoula, X., Siaterlis, G., Nikolakis, N. and Alexopoulos, K. (2021). A deep learning model for predictive maintenance in cyber-physical production systems using lstm autoencoders, *Sensors* **21**(3): 972.

Bouabdallaoui, Y., Lafhaj, Z., Yim, P., Ducoulombier, L. and Bennadji, B. (2021). Predictive maintenance in building facilities: A machine learning-based approach, *Sensors* **21**(4): 1044.

Cakir, M., Guvenc, M. A. and Mistikoglu, S. (2021). The experimental application of popular machine learning algorithms on predictive maintenance and the design of iiot based condition monitoring system, *Computers & Industrial Engineering* **151**: 106948.

Carvalho, T. P., Soares, F. A., Vita, R., Francisco, R. d. P., Basto, J. P. and Alcalá, S. G. (2019). A systematic literature review of machine learning methods applied to predictive maintenance, *Computers & Industrial Engineering* **137**: 106024.

Çınar, Z. M., Abdussalam Nuhu, A., Zeeshan, Q., Korhan, O., Asmael, M. and Safaei, B. (2020). Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0, *Sustainability* **12**(19): 8211.

Davari, N., Veloso, B., Ribeiro, R. P., Pereira, P. M. and Gama, J. (2021). Predictive maintenance based on anomaly detection using deep learning for air production unit in the railway industry, *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, IEEE, pp. 1–10.

Davari, Narjes, V. B. R. R. and Gama, J. (2023). MetroPT-3 Dataset, UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5VW3R.

Frankó, A., Hollósi, G., Ficzere, D. and Varga, P. (2022). Applied machine learning for iiot and smart production—methods to improve production quality, safety and sustainability, *Sensors* **22**(23): 9148.

Gatta, F., Giampaolo, F., Chiaro, D. and Piccialli, F. (2022). Predictive maintenance for offshore oil wells by means of deep learning features extraction, *Expert Systems* p. e13128.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory, *Neural computation* **9**(8): 1735–1780.

Liu, C., Tang, D., Zhu, H. and Nie, Q. (2021). A novel predictive maintenance method based on deep adversarial learning in the intelligent manufacturing system, *IEEE access* **9**: 49557–49575.

Martínez-Llop, P. G., Bobi, J. d. D. S. and Ortega, M. O. (2023). Time consideration in machine learning models for train comfort prediction using lstm networks, *Engineering Applications of Artificial Intelligence* **123**: 106303.

Namuduri, S., Narayanan, B. N., Davuluru, V. S. P., Burton, L. and Bhansali, S. (2020). Deep learning methods for sensor based predictive maintenance and future perspectives for electrochemical sensors, *Journal of The Electrochemical Society* **167**(3): 037552.

Nguyen, K. T. and Medjaher, K. (2019). A new dynamic predictive maintenance framework using deep learning for failure prognostics, *Reliability Engineering & System Safety* **188**: 251–262.

Schröer, C., Kruse, F. and Gómez, J. M. (2021). A systematic literature review on applying crisp-dm process model, *Procedia Computer Science* **181**: 526–534.

Schwendemann, S., Amjad, Z. and Sikora, A. (2021). A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines, *Computers in Industry* **125**: 103380.

Serradilla, O., Zugasti, E., Rodriguez, J. and Zurutuza, U. (2022). Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects, *Applied Intelligence* **52**(10): 10934–10964.

Silvestrin, L. P., Hoogendoorn, M. and Koole, G. (2019). A comparative study of state-of-the-art machine learning algorithms for predictive maintenance., *SSCI*, pp. 760–767.

Traini, E., Bruno, G., D'antonio, G. and Lombardi, F. (2019). Machine learning framework for predictive maintenance in milling, *IFAC-PapersOnLine* **52**(13): 177–182.

Veloso, B., Gama, J., Ribeiro, R. and Pereira, P. (2022). MetroPT2: A Benchmark dataset for predictive maintenance.
URL: *https://doi.org/10.5281/zenodo.7766691*

Veloso, B., Ribeiro, R. P., Gama, J. and Pereira, P. M. (2022). The metropt dataset for predictive maintenance, *Scientific Data* **9**(1): 764.

Wang, Y., Du, X., Lu, Z., Duan, Q. and Wu, J. (2022). Improved lstm-based time-series anomaly detection in rail transit operation environments, *IEEE Transactions on Industrial Informatics* **18**(12): 9027–9036.

Wu, H., Huang, A. and Sutherland, J. W. (2020). Avoiding environmental consequences of equipment failure via an lstm-based model for predictive maintenance, *Procedia Manufacturing* **43**: 666–673.

Zhang, W., Yang, D. and Wang, H. (2019). Data-driven methods for predictive maintenance of industrial equipment: A survey, *IEEE systems journal* **13**(3): 2213–2227.

Zhuang, L., Xu, A. and Wang, X.-L. (2023). A prognostic driven predictive maintenance framework based on bayesian deep learning, *Reliability Engineering & System Safety* **234**: 109181.