Predictive maintenance

Leonardo Anyelo Rodriguez Martinez
Dept. of Computer Science
Hof University of Applied Sciences
Hof, Germany
Leonardorm7@hotmail.com

***Abstract*—Predictive maintenance plays an important role in industries, as manufacturing companies often incur significant maintenance expenses and suffer financial losses due to unexpected machine failures. These failures lead to breakdowns that negatively impact overall company efficiency. Therefore, it is important to implement systems capable of predicting when machines are likely to fail in order to perform maintenance in advance and avoid costly defects and unplanned breakdowns. This paper aims to outline the necessary steps for implementing predictive maintenance, from data analysis to the application of AI models capable of predicting failures or estimating the remaining useful life (RUL) of machines. For this purpose, two datasets were utilized to explore two predictive maintenance approaches: condition monitoring and RUL estimation. The first dataset was analyzed for condition monitoring to detect machine failures, while the second focused on predicting the RUL of batteries. Various models were trained and their performance compared, demonstrating that in some scenarios, simple models can achieve performance comparable to or better than complex models. The code implemented for this study is available at the following link.**

***Index Terms*—Predictive maintenance, condition monitoring, remaining useful life, data analysis, AI models.**

I. Introduction

A common problem faced by manufacturing industries is the deterioration of the equipment which becomes critical when it results in expensive repairs or downtimes. Therefore, maintenance is essential in all industries to ensure that machines operate optimally, which results in increased production efficiency and product quality, since any unexpected downtime can cause a significant financial loss. According to [1], industries lose approximately 5% - 20% of their productive time in unplanned downtimes, representing a significant financial loss. To deal with this, various maintenance approaches are employed, such as reactive and preventive maintenance. However, the evolution of modern techniques has enabled the transition from reactive and preventive maintenance strategies to predictive maintenance.

Reactive maintenance consists of performing maintenance when the equipment fails. This approach results in high repair costs and unexpected downtimes that affect the production process. On the other hand, the majority of companies implement preventive maintenance, which consists of performing periodic maintenance to avoid failures and downtimes. However, this approach can lead to unnecessary maintenance, increasing the nonoperational time of the machine and overall maintenance costs [1]. Therefore, predictive maintenance aims to reduce maintenance frequency by performing it only when truly necessary, before the equipment fails.

The advancements in technology, particularly through the implementation of IoT systems, have facilitated the collection of data from sensors installed in machines. Combined with big data methodologies for preprocessing and analysis, along with advanced machine learning techniques, it is now possible to predict machine failures.

Predictive maintenance aims to anticipate failures in advance to prevent them. However, there are different approaches to predictive maintenance. For this study, health monitoring and predicting the Remaining Useful Life (RUL) will be analyzed, RUL consists of estimating the remaining time before a component or system fails. Condition Monitoring (Health Monitoring) involves tracking real-time data, such as temperature, vibration, or pressure to detect anomalies that might indicate a failure using non-invasive techniques.

Machine learning techniques play a crucial role in these approaches by training algorithms to analyze historical and real-time data to identify degradation patterns. The implementation of these trained models allows maintenance activities to be scheduled before failures occur, reducing downtime and maintenance costs while extending the equipment's lifespan.

For this reason, predictive maintenance has become a promising approach to reduce machine downtimes. Predictive maintenance involves collecting data over time from various machines and equipment. This data is then analyzed to identify correlations and patterns, which can help to predict and prevent potential machine failures, reducing costs associated with unexpected downtimes and maintenance expenses. [2] The main objectives of predictive maintenance focus on cost reduction and reliability improvement.

However, when implementing predictive maintenance in complex systems composed of multiple machines, each with distinct maintenance costs and degradation processes, a single optimization criterion such as minimizing maintenance cost, maximizing system reliability, or minimizing equipment downtime is not enough. Focusing on one objective, such as cost minimization, may lead to results that do not meet other essential objectives, like feasibility or reliability. Therefore, multi-objective optimization becomes essential, allowing for a balanced trade-off that takes into account the unique attributes of each machine to optimize maintenance across the system.

To address these challenges, different algorithms such as random forest, support vector machines (SVM), logistic regression, and neural network architectures, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long-Short-Term Memory (LSTM), and Deep Neural Networks (DNN), have been implemented. These models excel in understanding and detecting patterns in complex datasets, enabling predictive maintenance systems to identify potential failures and accurately predict the remaining useful life (RUL) of the equipment. By applying these neural network models, predictive maintenance can achieve the necessary balance between multiple objectives, improving system reliability and reducing maintenance costs.

This paper aims to compare different models using two datasets to evaluate their performance in two predictive maintenance approaches: condition monitoring to predict machine failures based on sensor signals and RUL prediction.

II. LITERATURE REVIEW

Maintenance is essential to ensure production efficiency and minimize costs caused by unexpected downtimes. For instance, the research [3] presents a digital twin prototype of a rotor system. This prototype consists of three components: the real machine, a digital representation of the machine (digital twin), and the data connection that synchronizes changes between the real machine and its digital counterpart. The study demonstrated good results in diagnosing rotor unbalance faults and predicting their progression, reducing the error to less than 5%. However, constructing digital twin models can be challenging in some cases due to the complex structures and varying operating conditions of real machinery.

While digital twin models have shown promise in diagnosing faults and predicting maintenance needs, the integration of advanced techniques, such as deep learning, can further enhance the monitoring and prediction capabilities of machinery health. Various studies have summarized deep learning techniques for machine health monitoring, such as Autoencoder (AE), Deep Belief Network (DBN), Deep Boltzmann Machines (DBM), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN) [4], [5]. For example, [2] discusses various predictive maintenance approaches, including classification models designed to identify early failures by classifying whether a machine is likely to fail soon, anomaly detection models capable of flagging unusual behaviors even without prior knowledge of specific failure patterns, providing further insight into unexpected issues, and regression models, which can predict the Remaining Useful Life (RUL) based on degradation processes learned from historical data.

As machine learning techniques continue to evolve, the advancement of big data techniques and the improvement of hardware like GPUs have enabled more complex mathematical operations to be performed faster than before, enhancing the performance of machine learning algorithms, which are a powerful solution for extracting important insights and making appropriate decisions based on big data. Machine learning techniques can be divided into two types: shallow and deep learning algorithms.

In the research [2] and [5], some traditional machine learning-based approaches are mentioned. Among the shallow techniques, the common algorithms mentioned are:

- **Artificial Neural Networks (ANNs)**: are algorithms inspired by how the human brain processes information and have gained importance in fault diagnosis and prognosis. For example, in the study [6], an ANN classifier was applied to vibration signals, achieving a fault classification accuracy rate of 97%. This demonstrates that ANNs can be effective for classification and prediction tasks. However, since ANNs contain many weight parameters that need to be trained, they require significant computational resources and are prone to overfitting.
- **Decision Tree (DT)** is a method used for classification and regression, aiming to predict the target variable's label or class by learning decision rules inferred from data features. C4.5 [7] is one of the most widely used algorithms for generating decision trees. DTs are easy to interpret and understand. However, they are prone to overfitting and may suffer from poor prediction accuracy in certain cases. Decision trees are typically applied in fault prognosis for turbofan engines, lithium batteries, and in fault diagnosis for rail vehicles, anti-friction bearings, and refrigerant flow systems.
- **Support Vector Machines (SVM)** are networks that work well with both unstructured and semi-structured data, can handle high-dimensional features, and have a lower risk of overfitting. However, SVMs are less efficient with large datasets and do not provide a probabilistic explanation for classifications. They have been used for fault diagnosis in rotating machinery, bearings, and wind turbines.
- Finally, research shows that **k-Nearest Neighbors (k-NN)** is an algorithm easy to implement. However, it is sensitive to unbalanced datasets and noisy attributes, which can affect the model's accuracy. It is commonly used for fault diagnosis, Remaining Useful Life (RUL) prediction, and early fault warning.

On the other hand, deep learning has shown a superior ability in feature learning, fault classification and fault prediction. The research [2] highlights the most common deep learning techniques:

- **Autoencoders (AEs)** are promising neural networks for learning high-level representations from raw data. They are used to compress complex data, which helps in pattern detection, fault diagnosis, and predicting the remaining useful life (RUL) of equipment. Autoencoders work by encoding the input data into a compressed format and then decoding it to reconstruct the original input, thereby helping to identify patterns and structures in the data. They can handle multi-sensory data and are often combined with classification or regression models. However, they require large datasets for pretraining and may struggle to determine which information is most relevant. Autoencoders are commonly used for feature extraction, multi-sensory data fusion, fault diagnosis, degradation process estimation, and RUL prediction.

- **Convolutional Neural Networks (CNNs)** are deep neural networks recognized for their weight-sharing property and local field representation ability, which enable them to capture and analyze complex details in data, such as signal patterns or unlabeled information. This makes CNNs widely applicable in fault diagnosis and tool life prediction. They can automatically extract health indicators without requiring prior expert knowledge, enhancing their ability to identify trends and anomalies in machine data. CNNs often outperform ANNs in tasks such as image recognition, with lower complexity and reduced memory usage. However, CNNs are prone to overfitting, have high computational costs, and require large amounts of training data. They are typically used in fault diagnosis, degradation process estimation, and RUL prediction.
- **Recurrent Neural Networks (RNNs)** are specialized for sequential data, such as temperature readings or machine vibrations over time. However, they often encounter challenges with vanishing and exploding gradients during backpropagation, which makes learning long-term dependencies difficult. To mitigate these issues, architectures like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) were developed. These advanced networks are particularly effective for fault diagnosis in complex systems and RUL prediction.
- **Deep Belief Networks (DBNs)** are deep learning architectures formed by stacking multiple Restricted Boltzmann Machines (RBMs), where the output of each layer serves as the input to the next. This layer-by-layer training allows DBNs to learn hierarchical features, and parameters can be adjusted using a softmax classification layer. DBNs have proven effective for fault diagnosis in complex systems.
- **Generative Adversarial Network (GAN)** consists of two networks, a generator and a discriminator. The generator creates synthetic samples (such as time series data), while the discriminator tries to distinguish these samples from real ones, allowing both models to improve over time. In predictive maintenance, GANs are useful for generating synthetic data to balance classes in unbalanced datasets. GANs can also be applied to anomaly detection by generating anomaly scores, which help in accurate failure detection. Additionally, GANs are used for failure prognosis by estimating remaining useful life through projections of future failure points based on historical data.

Nevertheless, industries also implement techniques such as Transfer Learning, which involves using a model that has already been pretrained on one domain and then fine-tuning it for a related, specific domain, thus improving accuracy. This approach reduces training time and leverages pretrained parameters to improve the precision of the model.

Moreover, a single method is often insufficient. Therefore, in some cases, hybrid approaches are used to overcome the limitations of relying on a single technique. For example, failure detection typically requires labeled data with previous failure instances. However, many datasets lack previous failure labels, making it challenging to apply conventional models for fault detection. To address this, [8] proposes a deep learning method for anomaly detection without labeled data. This approach combines two types of neural networks: stacked autoencoders (SAE), which learn key features in an unsupervised manner, and long short-term memory (LSTM) networks, which detect potential anomalies in the data. This method achieved 99% accuracy in identifying anomalies without labeled data.

Another example of a hybrid approach is presented in [9], where the authors implement CNNs and LSTMs to improve remaining useful life (RUL) predictions. In this approach, an LSTM and a CNN operate in parallel, with their outputs combined and then fed into an additional LSTM to generate the final RUL prediction.

While these methods focus on anomaly detection and failure prediction, performing predictive maintenance involves several steps. In [5], the authors summarize the predictive maintenance stages required to meet industrial requirements. The first step, after data collection, is the preprocessing stage. This involves preparing the data using common techniques such as sensor data validation, which ensures the data is accurate; feature synchronization to align signals sampled at different timestamps, creating time-series data that is easier to handle; data cleaning to remove or interpolate missing values; oversampling to balance imbalanced datasets; encoding and discretization to transform the data into a format suitable for processing; segmentation to divide the data into chunks for parallelization; normalization or standardization to scale all features to a similar range; and noise handling to manage noisy data.

After preprocessing, the next step is feature engineering, which involves extracting a relevant subset of features to be used as input for statistical and machine learning models. However, this step may not be necessary for deep learning models, as they can automatically extract features. Common techniques in feature engineering include feature extraction and projection to a new space to reduce dimensionality while preserving relevant information, concatenation and fusion to create new features by combining the existing ones, and feature selection to discard features with low variance, redundancy, or lack of correlation with the target variable.

The following stage is anomaly detection, which aims to determine if the machine is operating under normal conditions. Methods implemented typically include classification and clustering. This is followed by the diagnosis step, which identifies whether the detected anomaly indicates a failure. Diagnosis is usually based on root cause analysis (RCA) to identify the cause of the problem. The prognosis stage then estimates the remaining useful life based on historical data. Finally, the mitigation stage involves performing maintenance actions before the predicted failure occurs to prevent downtime and restore the equipment to optimal working condition.

To illustrate how these stages are applied in practice, an implementation of machine learning in predictive maintenance is demonstrated in [1]. The authors developed a web application that takes sensor data as input to predict potential machine downtime. This process involves data collection, cleaning, and training a suitable machine learning model to predict system failures. The paper details the use of an LSTM model to make predictions based on data collected over time from sensors in a tubing machine. The model predicts machine failure by analyzing parameters such as temperature, pressure, machine speed, revolutions per minute, and material input.

III. METHODOLOGY

This section describes the process used to analyze datasets for health monitoring and remaining useful life (RUL) estimation. It provides a detailed description of the datasets, the data analysis techniques employed, and the models implemented for predicting failures and RUL. The subsequent section will discuss the results obtained after training the models on each dataset.

A. Datasets description

To explore two different approaches for predictive maintenance, two datasets were implemented: one for health monitoring and another for remaining useful life prediction.

1) Dataset for health monitoring: For this approach, a public dataset [10] was selected. This dataset contains 944 values collected from various machines to predict failures and includes 10 features:

- **Footfall:** Represents the number of people or objects near the machine. This feature may indicate whether the machine is located in a high activity area where external factors could influence its performance.
- **TempMode:** Denotes the machine's temperature configuration. It could reveal if specific temperature settings impact performance or contribute to faults.
- **AQ (Air Quality Index):** Measures the air quality around the machine, as contaminated air could contribute to premature wear.
- **USS (Ultrasonic Sensor):** Provides proximity measurements, detecting nearby objects.
- **CS (Current Sensor):** Measures the machine's electrical consumption, which could be indicative of operational anomalies.
- **VOC (Volatile Organic Compounds):** Measures the concentration of volatile organic compounds near the machine, as high levels may indicate the presence of chemicals that could cause corrosion or failures.
- **RP (Rotational Position):** Indicates revolutions per minute of the machine, which is useful for analyzing performance.
- **IP (Input Pressure):** Indicates the input pressure of the machine.
- **Temperature:** Reflects the operating temperature of the machine, which may signal overheating or cooling system issues.
- **Fail:** A binary indicator of machine failure (1 indicates failure, 0 indicates normal operation). This is the target variable for the models, which will learn to predict machine failures based on the other features.

2) Dataset for remaining useful life prediction: For this approach, the public dataset [11] was implemented for RUL predictions of lithium-ion batteries. This dataset captures the aging process of various batteries and includes the following features:

- **Capacity:** Evaluates the degradation of the battery.
- **Internal resistance (RE):** Represents the electrical resistance of the battery.
- **Chrge transfer resistance (Rct):** Indicates charge movement efficiency.
- **Ambient temperature:** Reflects the external temperature, which affects battery performance.
- **Battery ID:** Identifies each battery.
- **Test ID:** Links specific test conditions to outcomes.
- **UID and Filename:** Serve as traceable dataset references.

The data were collected through repeated charge and discharge cycles. The experiments concluded when the batteries reached the end of their useful life.

B. Data analysis

1) Data analysis for health monitoring: The code for the process detailed in this section can be found in this GitHub repository. Once the data is collected, the first step in analyzing it is to check for missing values. However, no missing values were identified in this dataset. Next, the dataset must be analyzed to determine whether the target values are balanced. In this case, the dataset contains 551 instances of the target value 0 and 393 instances of the target value 1. This indicates that 42% of the target values represent failures, suggesting a moderate imbalance. Therefore, when splitting the dataset, it is essential to ensure that the target values are evenly distributed between the training and testing datasets.

By analyzing the data Fig. 1 and Fig. 2, it can be observed that the feature ranges vary significantly. Therefore, the data needs to be scaled to maintain consistent ranges, as well as uniform minimum and maximum values across all features, to facilitate proper processing.

Before scaling the data, a correlation matrix Fig. 3 was generated to observe the correlation between the different sensors and the failure.

According to the correlation matrix, the most significant variables related to failures are the air quality index (AQ) and the volatile organic compounds (VOC), both of which represent a strong positive correlation with failures. In contrast, the ultrasonic sensor (USS) demonstrates a negative correlation with failures. From this analysis, it can be inferred that volatile

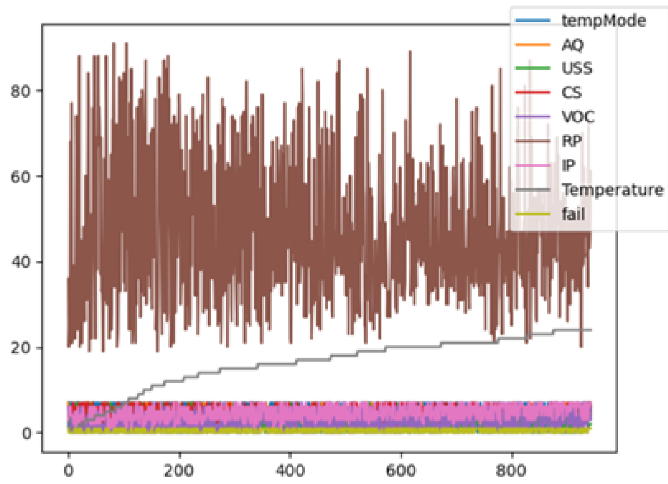

Fig. 1. Visualization of Dataset Features

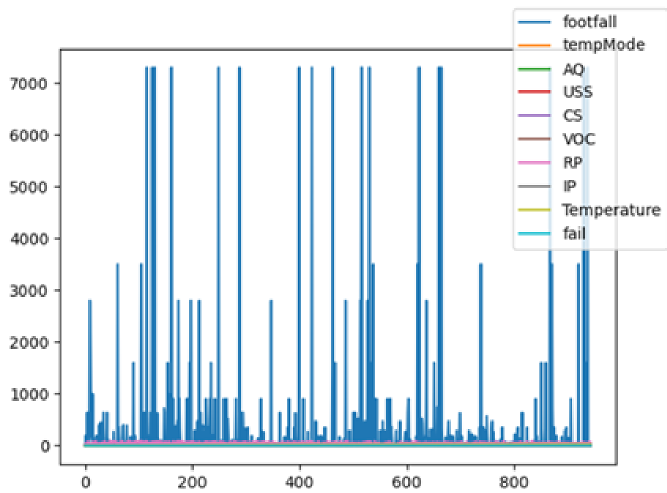

Fig. 2. Graph of Dataset Features, Including 'Footfall'

organic compounds might contribute to failures, while higher ultrasonic sensor values decrease the probability of failures. This might indicate that the sensor measures objects near the machine, and the absence of nearby objects might correlate with reduced levels of volatile organic compounds, thereby reducing the probability of failures.

Following this analysis, an outlier analysis was conducted to identify outliers in the dataset, as these could affect the scaling process. In this case, the dataset contains several outliers: the 'footfall' feature contains 154 outliers, the 'current sensor' feature contains 87 outliers, and the 'temperature' feature has 48 outliers. Considering that both temperature and current can vary depending on machine performance, these values might indicate transitional states, such as the machine starting up or shutting down.

To assess the impact of the outliers, models were trained with both the outliers and without the outliers. This resulted in better performance when the outliers were removed. Therefore, further analysis was conducted after removing the outliers. Additionally, the 'footfall' feature, which represents the number of people near the machine, was considered less relevant to machine behavior. However, two analyses were performed: one including the 'footfall' feature and one excluding it. The models trained without the 'footfall' feature demonstrated slightly better performance. Therefore, this feature was removed along with the identified outliers.

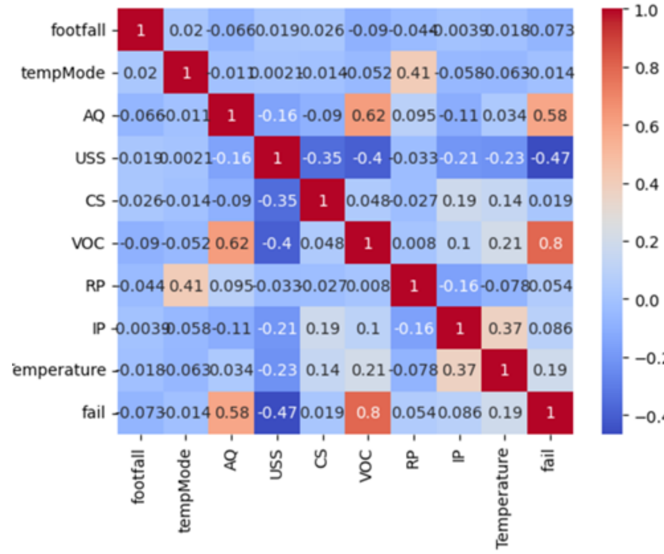

Fig. 3. Correlation matrix

Once the data was cleaned, the dataset was split into training and testing sets. The training data, as the name suggests, is used to train the models, while the testing data is used to evaluate their performance in predicting failures. Two variables were created: 'x', containing the eight features, and 'y', containing the target values (the 'fail' column).

This is an important step because scaling and Principal Component Analysis (PCA) must be performed on the training data only. This ensures that the testing data remains unseen during these processes, preventing data leakage. The data set was divided into training (80%) and testing (20%) datasets, using the stratify parameter to maintain the same proportion of the target values (fail) in both datasets. As a result, four variables were obtained: x_train, x_test, y_train, and y_test.

After splitting, the data was scaled. The scaler was fitted exclusively on the training data to avoid contamination from the testing data. Both training and testing data were then scaled, resulting in all features having values between 0 (minimum) and 1 (maximum).

After scaling, PCA was applied to reduce the dimensionality of the dataset. To determine the minimum number of components needed to retain most of the original information, an explained variance ratio plot was generated Fig. 4. This analysis revealed that seven components were required to retain at least 95% of the information.

Therefore, PCA was performed, retaining 7 components, and four different models were trained using the transformed data.

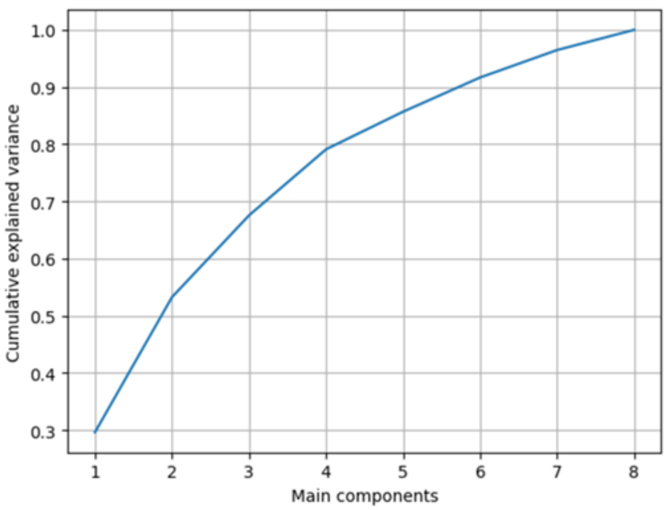

Fig. 4. Explained variance ratio plot

2) Data analysis for remaining useful life prediction: For this dataset, the target values (RUL) were created by grouping the data for each battery based on its battery ID. A value of 1 was assigned to the first entry, which represents the full remaining life, and this value gradually decreases to 0 for the last value, which represents the end of the battery's life. The code for this analysis can be found in this GitHub repository. After creating the target values, data analysis was performed similarly to the previous dataset. Therefore, this process will not be explained in detail. However, key steps included checking for missing values and identifying the useful features for the analysis. In this case, it was found that the useful features are ambient temperature, capacity, Re, and Rct. These four features were analyzed, while the others were deleted as they referred to other files not relevant for the analysis. Additionally, outliers were also removed from the dataset, with a total of 205 values deleted.

Once the data was cleaned and analyzed, the dataset was divided into training and testing data. The four features and the target values (RUL column) were split using a random split, with 80% of the data for training and 20% for testing, resulting in x_train, x_test, y_train, and y_test.

After splitting the data, the features, which have different ranges, were scaled in the same way as for the previous dataset. The scaler was fitted using the training data, and then the data was scaled. In contrast to the previous dataset, where dimensionality reduction was necessary, this dataset contains only four features, so PCA was not required. Finally, four different models were trained using this processed data.

C. Model Selection

For model selection, considering that there are two different approaches, classification and regression, baselines were first defined to determine the minimum viable algorithms. For classification, the baseline is an algorithm that always predicts the most frequent class, and for regression, it is an algorithm that always predicts the mean of the values. These baselines define the threshold, meaning that the implemented models should perform better than these simple algorithms. The metrics used to evaluate the performance of the models were accuracy for classification, which measures the proportion of the correctly predicted values out of the total number of values, and Mean Squared Error (MSE) for regression, which calculates the average of the squared differences between the model predictions and the real values.

1) Classification approach: In this case, the models need to learn from the data collected by the sensors to determine whether the machine fails or not. This is a binary classification problem, where the goal is to classify the failure class. Therefore, the algorithms to implemented in this case will focus on binary classification tasks. For this task, two lineal models were implemented: Logistic Regression and Support Vector Machine (SVM), a Random Forest model which uses multiple decision trees to improve generalization and reduce overfitting, and a Deep Neural Network (DNN) to compare the performance among the models.

To decide the best model, the baseline performance was determined using a model that always predicts the most frequent target value. In this case, the dataset contains 447 instances of the failure class labeled 0 and 370 instances labeled as 1. This means that a model that always predicts the target value 0 would achieve 58% of accuracy by correctly predicting the value 0 in 58% of the cases. Taking this into account, the selected model should achieve better performance than 58%.

2) Regression approach: In this case, the models need to learn from the data collected to determine the remaining useful life (RUL) of the batteries. This is a regression problem, where the goal is to predict a continuous value representing the remaining useful life of the battery. Therefore, the implemented models focus on regression tasks. For this task, the following models were implemented: a Random Forest Regressor, a Support vector machine, a K-Neighbors Regressor and a Deep Neural Network (DNN).

To determine the baseline performance of a basic model, an algorithm that always predicts the mean was used. For the dataset used in this approach, the mean squared error (MSE) of a model that always predicts the mean value (0.5) is 0.086. Therefore, the implemented models should achieve a lower error than this baseline.

IV. RESULTS

This section presents the results obtained after training different machine learning methods to predict machines failures and the remaining useful life of batteries.

1) Classification approach: After training the models, the accuracy scores shown in Table I were obtained.

In general, it can be seen that the four models achieved good accuracy. However, the best accuracy was obtained for the Deep Neural Network (DNN). The DNN implemented consists of three hidden layers with 128, 64 and 32 neurons, using

	Logistic Regression	SVM	Random Forest	Deep Neural Network
Accuracy	0.95	0.94	0.92	**0.96**

TABLE I
ACCURACY SCORES OF DIFFERENT CLASSIFICATION MODELS

ReLU as the activation function and one output layer with Sigmoid as the activation function to obtain the probability between 0 and 1. Adam was used as the optimizer with a learning rate of 0.001, and the model was trained for 100 epochs.

Although the DNN achieved the best accuracy, it has some drawbacks. DNNs are difficult to explain, require high computational power, and often have complex structures. Moreover, since the Logistic Regression model achieved nearly identical performance, it would be the best model to implement in a real scenario. Logistic Regression is simpler, easier to implement, more explainable, and faster to train. Therefore, despite the slightly better performance of the DNN, the Logistic Regression model would be the best choice for real-world implementation due to the disadvantages of the DNN in this context. Furthermore, it also achieved good performance. As shown in Fig. 5, the confusion matrix illustrates the number of correct and incorrect predictions made by the Logistic Regression model.

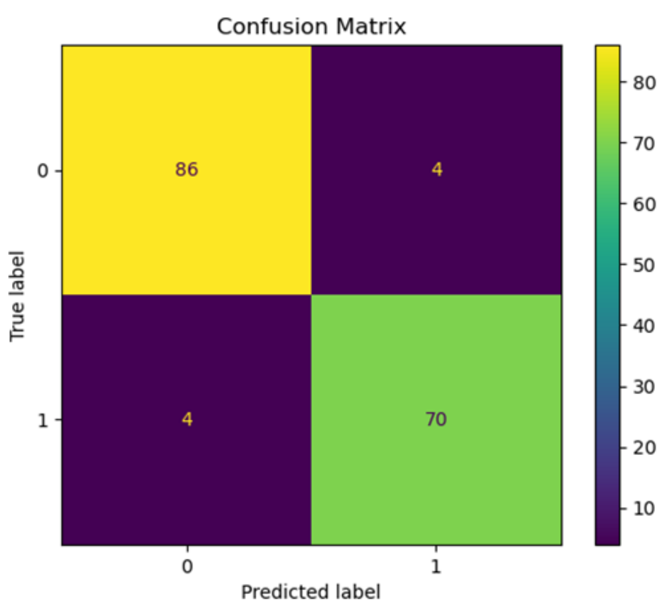

Fig. 5. Logistic regression confusion matrix

After training the models, they can identify when a machine is failing. This is particularly useful for health monitoring in closed environments where it is not possible to visually identify machine failures. By using a model to monitor the machine, failures can be detected as they occur.

However, predicting failures in advance would be even more beneficial, as it would allow immediate action to be taken, preventing further damage or severe breakdowns.

Ideally, the model would predict failures before they occur. Unfortunately, the dataset used lacks timestamps or data indicating the number of cycles a machine has undergone before failure. As a result, the models could not be trained to predict failures in advance. To address this limitation, some trials were conducted where the models were trained to detect failures five measurements before the actual failure event.

This was achieved by shifting the 'fail' column five measurements earlier, creating a moving window. The size of the window can be adjusted based on specific requirements for timely failure detection, such as how many measurements earlier the model could predict the failure. However, this approach did not work well for this dataset, as the sensor data showed no correlation with future failures Fig. 6. As a result, the models struggled to learn meaningful patterns for predicting future failures, and their performance dropped to around 61%.

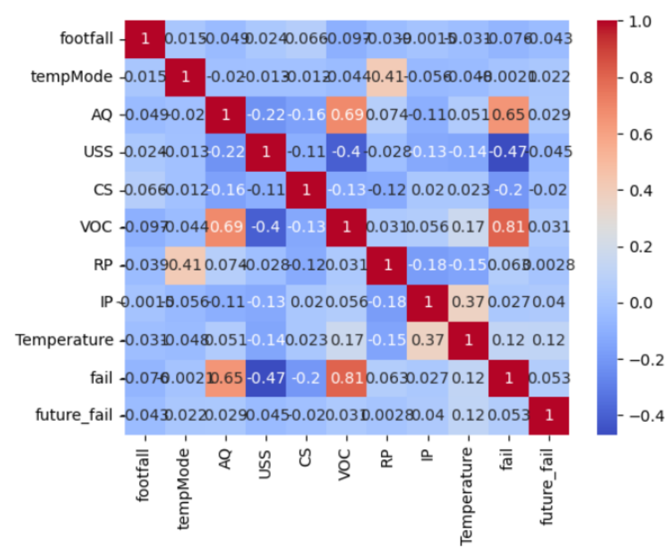

Fig. 6. Correlation matrix for future failure prediction

Additionally, attempts to predict future values using this dataset, including a Remaining Useful Life (RUL) analysis, also yielded inadequate results. Therefore, this dataset is not suitable for such approaches and is better suited for health monitoring rather than future failure prediction, as it contains failures from different machines and lacks enough data to predict future failures or the RUL of a specific machine.

Once the model is selected for the health monitoring approach, the final step is its implementation in the real process with real-time data. To do this, the data must have the same structure as the training data. Then, the data must be scaled, and the PCA must be applied using the scaler and the PCA parameters adjusted with the training data, before passing the real-time data to the model.

Once the preprocessed real-time data from the machines is provided to the model, it will generate a prediction for each group of signals transmitted. This is useful for health monitoring, generating alarms when the machines are failing to prevent serious failures.

2) Regression approach: After training the models, the following Mean square error (MSE) scores were obtained:

	Random Forest Regressor	Support Vector	KNeighbors Regressor	Deep Neural Network
MSE	**0.005**	0.04	0.013	0.02

TABLE II
COMPARISON OF MSE BETWEEN MODELS

In this case, all four models performed better than the baseline. However, the best model was the Random Forest Regressor, which uses 100 trees and achieved an MSE of 0.005. Fig. 7 shows the results for this model, comparing the real values to the predictions. In this case, it can be seen that a simpler algorithm outperformed a DNN. In this study, the DNN was tested with various hyperparameters, and the results shown in Table II were obtained using a DNN with layers containing 256, 128, 64, and 32 neurons, a ReLU activation function, and the Adam optimizer with a learning rate of 0.001, and trained for 200 epochs. This demonstrates that, in some cases, basic models can outperform more complex ones, such as the DNN. It is also worth noting that adjusting the architecture of the DNN could improve its performance. This highlights the complexity of DNN models compared to simpler algorithms. However, considering the performance achieved by the Random Forest Regressor and its simplicity, even if the DNN were to achieve better performance, it would still be reasonable to use the simpler model for the reasons explained previously.

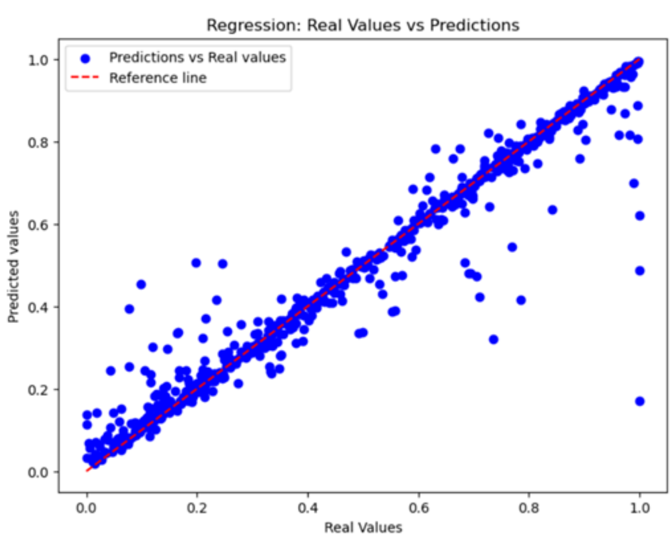

Fig. 7. Regression results showing real values vs predictions

Finally, for the implementation, it is necessary to take the real-time data in the same format as the training data and scale it using the scaler fitted with the training data. After this, the data is given to the model, which will predict the remaining useful life as a value in a range from 0 to 1. A threshold, for example, can be set to 0.5. Therefore, when the batteries fall below 50% of their RUL, an alarm can be triggered based on how early it is needed to send the alarm for maintenance.

V. CONCLUSION

Predictive maintenance plays a crucial role in industries by predicting failures in advance, helping to avoid downtimes and reducing potential high repair costs. Based on the data collected and the required approaches, various neural network architectures can be applied, ranging from simple models to more complex ones, to extract anomaly indicators from complex systems.

This research reviewed the process of analyzing and training models for two different predictive maintenance approaches. Several models were trained, yielding good results in both approaches. For health monitoring, which involves analyzing patterns that might indicate failure, this approach is useful when machines are enclosed or when the failure is not immediately visible. For this approach, the best accuracy was obtained with a Deep Neural Network (96%). However, the Logistic Regression model also achieved good accuracy (95%). For remaining useful life prediction, the models can predict the remaining useful life of a machine and generate alerts before failure, allowing to perform maintenance to prevent breakdowns. For this approach, the best result was obtained by the Random Forest Regressor, with an MSE of 0.005.

Furthermore, it was demonstrated that depending on the type of problem and the data available, different predictive maintenance approaches can be implemented effectively. However, it is crucial to collect as much information as possible to fully understand the data, as each approach requires specific data.

Additionally, various algorithms for classification and regression problems were explored, showing their performance for the different approaches. It was also analyzed that, in some cases, it is important to evaluate whether the performance of a complex model justifies its use, or if a simpler model with slightly lower accuracy might be more practical and cost-effective in a real scenario.

VI. LIMITATIONS

The research was initially designed to focus only on the first dataset for health monitoring, implementing different approaches. However, when attempting to predict failures in advance, this was not possible due to the lack of sufficient information in the dataset. The models showed no correlation when shifting the target values to predict future failures. This was because the dataset contains failures from different machines, rather than a single machine, which complicates the analysis. Furthermore, the dataset does not contain enough data to support other predictive maintenance approaches. As a result, an alternative dataset was implemented to predict the remaining useful life of batteries.

VII. ACKNOWLEDGEMENT

I would like to acknowledge Prof. Dr. Christian Groth and Prof. Dr. Valentin Plenk for their support and guidance, which significantly contributed to the development of this project.

REFERENCES

[1] A. P. Kane, A. S. Kore, A. N. Khandale, S. S. Nigade, and P. P. Joshi, "Predictive maintenance using machine learning," *arXiv preprint arXiv:2205.09402*, 2022.

[2] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, "A survey of predictive maintenance: Systems, purposes and approaches," *arXiv preprint arXiv:1912.07383*, pp. 1–36, 2019.

[3] J. Wang, L. Ye, R. X. Gao, C. Li, and L. Zhang, "Digital twin for rotating machinery fault diagnosis in smart manufacturing," *International Journal of Production Research*, vol. 57, no. 12, pp. 3920–3934, 2019.

[4] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring," *Mechanical Systems and Signal Processing*, vol. 115, pp. 213–237, 2019.

[5] O. Serradilla, E. Zugasti, J. Rodriguez, and U. Zurutuza, "Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects," *Applied Intelligence*, vol. 52, no. 10, pp. 10 934–10 964, 2022.

[6] D. Pandya, S. Upadhyay, and S. Harsha, "Ann based fault diagnosis of rolling element bearing using time-frequency domain feature," *International Journal of Engineering Science and Technology*, vol. 4, no. 6, pp. 2878–2886, 2012.

[7] Z. Xiaoliang, Y. Hongcan, W. Jian, and W. Shangzhuo, "Research and application of the improved algorithm c4. 5 on decision tree," in *2009 International Conference on Test and Measurement*, vol. 2. IEEE, 2009, pp. 184–187.

[8] Z. Li, J. Li, Y. Wang, and K. Wang, "A deep learning approach for anomaly detection based on sae and lstm in mechanical equipment," *The International Journal of Advanced Manufacturing Technology*, vol. 103, pp. 499–510, 2019.

[9] J. Li, X. Li, and D. He, "A directed acyclic graph network combined with cnn and lstm for remaining useful life prediction," *IEEE Access*, vol. 7, pp. 75 464–75 475, 2019.

[10] U. Naeem, "Machine failure prediction using sensor data," https://www.kaggle.com/datasets/umerrtx/machine-failure-prediction-using-sensor-data, 2024, accessed: 2024-11.

[11] G. MANI, "Nasa battery data cleaned," https://www.kaggle.com/datasets/mystifoe77/nasa-battery-data-cleaned/data, 2024, accessed: 2024-12.