National College of Ireland

Predicting Electric Vehicle Failures Using Machine Learning: A Comparative Study on Imbalanced Sensor Data

MSc Research Project
Practicum 2

Alexis R. Santa Cruz Crespo
Student ID: x23277483

School of Computing
National College of Ireland

Supervisor: Faithful Onwuegbuche

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Alexis R. Santa Cruz Crespo		
Student ID:	23277483		
Programme:	MSc AI for Business	**Year:**	2025
Module:	Practicum 2		
Supervisor:	Faithful Onwuegbuche		
Submission Due Date:	15/09/2025		
Project Title:	Predicting Electric Vehicle Failures Using Machine Learning: A Comparative Study on Imbalanced Sensor Data		
Word Count:	6765	**Page Count** 24	

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: Alexis R. Santa Cruz Crespo

Date: 14/09/2025

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):	

Predicting Electric Vehicle Failures Using Machine Learning: A Comparative Study on Imbalanced Sensor Data

Alexis R. Santa Cruz Crespo
23277483
GitHub: https://github.com/AlexisSCC/Predictive_Maintenance_Project

Abstract

The use of Electric vehicles (EVs) is increasing but some unexpected technical failures can affect safety, reliability and operating costs. Traditional maintenance methods are often not able to detect early failures and it creates the need for predictive maintenance strategies. There are some limitations for machine learning models in EVs when they are working with imbalanced data because it can affect the performance. This study evaluates and compares four machine learning models: Logistic Regression, Decision Tree, Random Forest and XGBoost to predict Diagnostic Trouble Codes (DTC) using EV sensor data.

The project also studies the use of Synthetic Minority Oversampling Technique (SMOTE) to improve the performance of the model. The dataset has 131,396 units from three different profiles of EV users (rare, moderate and heavy), the dataset was cleaned, merged and processed. The models were trained using both datasets, imbalanced and balanced and they were evaluated using different evaluation metrics such as Accuracy, Balanced Accuracy, Precision, Recall, F1 Score, and ROC AUC. The results showed that without balancing the data, Logistic Regression could not detect failures and the tree-based models got high accuracy with a possible overfitting. After applying SMOTE, all the models improved their performance. Random Forest and XGBoost showed the best results between precision and recall and these results confirm that addressing class imbalance is very important for predictive failures in EVs and support the use of machine learning models for predictive maintenance to reduce downtime and costs. In the future we can study deep learning models with real-time data.

Keywords: Electric Vehicles, Predictive Maintenance, Machine Learning, SMOTE.

1 Introduction

Electric vehicles are more popular now because they help to protect the environment (Surabhi, 2024). However, they can be affected by unexpected technical failures, which can create problems of safety, downtime of the service and increase the repairing costs (Zhang et al., 2021). This situation has created a good opportunity to work in predictive maintenance methods to detect early signs of problems with the use of data information (Nunes et al., 2023).

Predictive maintenance is totally necessary because it can help to reduce costs and improve the trust of people on EVs (Theissler et al., 2021). Some studies explain that machine learning models like Random Forest works well predicting failures in different industries (Assagaf et al., 2023). Even when the use of ML models in electric vehicles is limited, some recent researches like Mishra et al., (2024) highlights the effectiveness of classification model to identify early failures. Also, Attia and Aoulmia (2025) tested different models such as Logistic Regression, Decision Tress and Random Forest using real time sensor data and they got an accuracy of 96%.

This project focuses in exploring how well artificial intelligence and different machine learning models can predict diagnostic trouble codes (DTC) in EVs to support more efficient maintenance decisions in the industry. The main objective is to compare and evaluate the performance of four ML in predictive maintenance. Some of the research questions that guide this project are:

- RQ1: How accurate are ML models to predict DTCs in different usage profiles?
- RQ2: Which are the variables that contribute most towards the predictions?
- RQ3: Are balancing techniques a good option to improve the performance of the model?

In this project two limitations are that the analysis focuses on binary classification of DTC and we are not working to identify the type of failures and the project assumes that the available data reflects important EV patterns and that they can be captured by machine learning models. The study is organized in six sections: Section 1 introduces to the research problem. Section 2 studies the related work in predictive maintenance. Section 3 shows the dataset and methodology. Section 4 describes the models and evaluation metrics. Section 5 shows the results and Section 6 are the conclusions and future work.

2 Related Work

Predictive Maintenance works preventing failures in different type of equipments. In electric vehicles where systems like batteries, electric motors and electronic units are very important, PdM is very useful and necessary (Mishra et al., 2021). Unlike traditional maintenance strategies, PdM is focus on the real condition of the components which is good to reduce downtime and extend the vehicle life. (Jauhar, S et al., 2024).

The EVs generate sensor datasets that could be useful to train machine learning models to prevent failures. Different studies demonstrated that predictive maintenance systems improve safety, reduce operative costs and help to monitor systems like batteries and power electronics (Zhang, X et al., 2021). However, there are some challenges like the quality of the data, class imbalance and the correct selection of evaluation metrics to measure these models (Nunes, P et al., 2023).

2.1 Machine Learning Models for Predictive Maintenance

Many machine learning models have been applied to predictive maintenance tasks in electric vehicles and they got good results. Some of the most important are the tree-based models such as Decision Tree (DT), Random Forest (RF) and XGBoost (XGB), which are often preferred because they are easy to interpret when working with structured data (Lou J, 2024). Random Forest showed that is a strong model to handle high dimensional and noisy data, which makes it suitable for electric vehicle diagnostics (Attia, R et al., 2025). XGBoost offers

higher accuracy and computational efficiency especially to work with imbalanced data or rare fault types (Tsylenko, M et al., 2022).

Although, models like Logistic Regression (LR) are simple, it still can be used because of its ability to provide probabilistic results and explainable decisions (Aydin, C et al., 2025). Some studies that compare these models suggest that three-based models often perform better than linear models in predictive maintenance tasks, especially when the variables are complex (Jafari, S et al., 2023). However, the final choice of the model depends on the data, computational limitations and specific objective of prediction.

2.2 Data Challenges in Predictive Maintenance

Feature engineering is a very important step in predictive maintenance because it transforms raw data in informative variables that improve the model learning. For electric vehicles, important variables include battery voltage, temperature, current and real-time data from OBD systems. Studies showed that a combination of sensor sources and derivative variables improves the accuracy of the model (Theissler et al., 2021). The use of vibration and voltage data is also very helpful to detect and prevent early failures. Also, feature selection methods such as correlation filtering and mutual information can help to reduce the noise and improve the interpretability (Prytz. R, 2014).

Class imbalance in failure prediction is a big problem, it happens when no failures are more common than the real failures. This imbalance makes the model to predict more the majority class and it means poor recall for failure detection. Oversampling and similar resampling techniques work well to improve the performance in those cases (Mahale, Y et al., 2025). Ignoring imbalance often means in low accuracy and bad results (Buabeng, A et al., 2021). Another evaluation metrics such as balanced accuracy or F1 score are recommended to obtain better information about the performance of the model (Theissler et al., 2021). In this project, class imbalance was addressed applying SMOTE techniques.

2.3 Evaluation Metrics

The correct selection of evaluation metrics is important to measure the performance of predictive maintenance models, especially in classification tasks. However, for example, is more common to use accuracy but the results can be wrong when we are working with imbalanced data, because they show the majority class and are not able to make a real prediction. To address this problem, there are more metrics such as precision, recall and F1 Score, which are also used to evaluate how well a model, detects failure cases. Some studies showec that depending only in accuracy may not get important information in models that fail identifying important variables (Spiegel, S et al., 2018).

The F1 score is a good metric for balancing false positives and false negatives, which can improve predictions and avoid breakdowns. Also, metrics like balanced accuracy and the area under the ROC curve AUC offer better information of the performance of the model with class imbalance. A good measure is balanced accuracy since it balances both classes (Achariyee, B et al., 2021). The ROC AUC measures the capability to separate the failure and no failure classes in classification models. These measures are used to make comparisons and know which machine learning model made a superior performance when predicting failures from the data provided on electric vehicles.

2.4 Research Gaps and Future Directions

Even though machine learning is very used in predictive maintenance, there are still some

limitations, especially when we are working with electric vehicles. Most of the studies focus on general industrial machines and not in specific problems to EVs, like battery or OBD system failures (Surabhi, 2024). Also, it is hard to find good datasets for EV maintenance, which makes it difficult to compare models (Chukwudi, I et al., 2024). Some models like deep learning need large amounts of data and a strong computing power, which is not always available (Kumar, A et al. 2023). Another problem is that many datasets have more no failure cases than failure ones and not all the studies deal with this imbalance in a proper way (Amihai, I et al., 2018). There are also few comparisons between classic models like logistic regression or decision trees for EVs (Benhanifia, A et al., 2025).

In conclusion, the literature shows that predictive maintenance using machine learning is still developing, but there are some important gaps, especially for electric vehicle applications. Most of the studies agree on the value of early fault detection and the use of machine learning models, but problems like class imbalance, limited EV datasets. This project aims to address these areas applying classic machine learning models to EV failure data, selecting useful variables, correcting data imbalance and evaluating the performance with different metrics.

3 Research Methodology

3.1 Dataset Overview

The dataset that we use in this study contains historical data that was collected from electric vehicles (EVs) under three types of users: Rare, which use their vehicle less frequent and only for short distances; Moderate, which use the vehicle on a daily basis, sometimes for long distances and Heavy, which have a more frequent usage, it includes coomercial and city drivings. We collected the data from Kaggle (Kunal Mehra, 2025), The data is from January 2020 to December 2024:

The original dataset included four user profiles but in this study only three were selected for analysis (Rare, Moderate, Heavy), the Regular user data was not included to maintain consistency and avoid redundancy. After merging the three individual CSV files, we created a column called user type to show the usage profile for each one. The dataset has different sensor variables like SOC (State of Charge), SOH (State of Health), Battery Temp, Motor Temp, Motor RPM, Motor Torque, Charging Cycles, Charging Voltage, Tire Pressure, Brake Pad Wear, etc.

Also, we transformed the main variable DTC (Diagnostic Trouble Code) to binary classification. The original variables were codes like POMR, POB, etc, and we cleaned and transformed it in a binary variable: No Failure = 0, Failure = 1. After cleaning the final dataset contained 131,396 rows for analysis and model development.

3.2 Exploratory Data Analysis (EDA)

The exploratory Data Analysis (EDA) was made to understand the characteristics of the dataset before building the predictive models. This process included analysis using some visualization.

3.2.1 Distribution of Numerical Variables

To understand the spread and shape of the dataset, we generate histograms for the most important numerical variables. These graphs show the variable distribution.

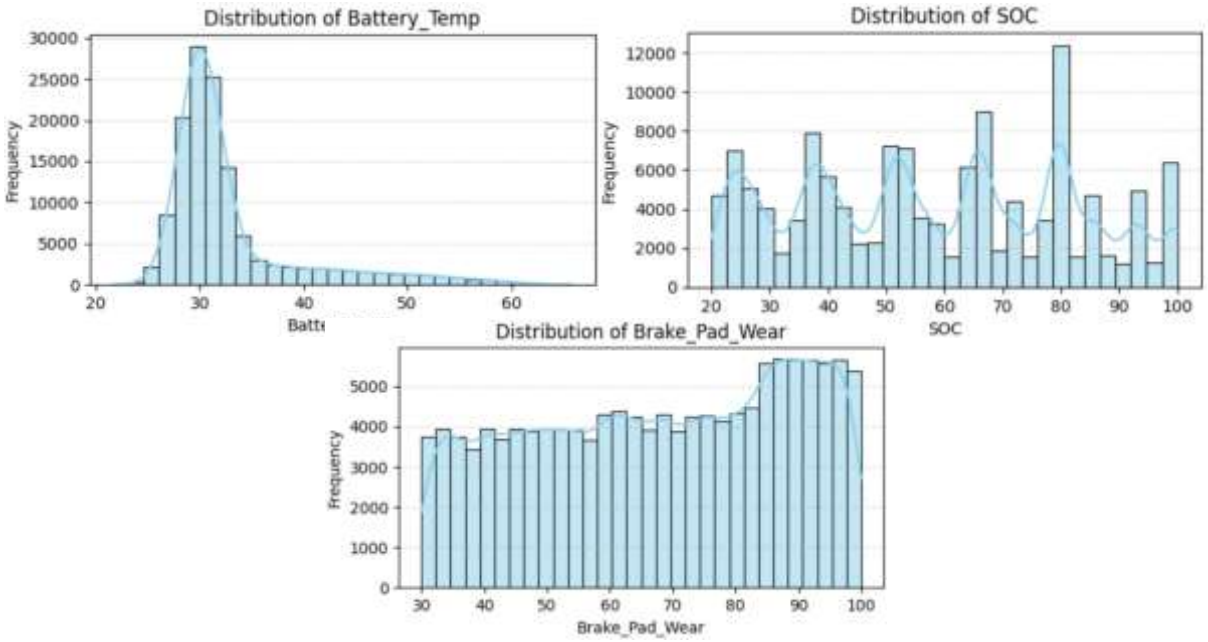

Figure 1: Distributions of Numerical Features

In the graph Fig. 1 we can see the behavior of some important variables, for example Battery Temp shows a slightly skewed distribution and most of the values are between 25 and 45°C, it could be because of an overload in the system. The SOC present a multimodal distribution with peaks in different charge levels. This suggests that every user has different charging habits, which can influence in the health of the battery and the vehicle in general. The distribution of Brake Pad Wear is wide and spread with many entries with more than 70%. This suggests many vehicles may be working under high bake usage, which can increase wear failure and maybe preventive maintenance.

3.2.2 Outlier Detection (Boxplots)

The boxplots were used to show the presence of outliers in the numerical variables. This step is very important because extreme values can affect the performance of the model and reduce the reliability in the predictions. In this case the three most affected variables were:

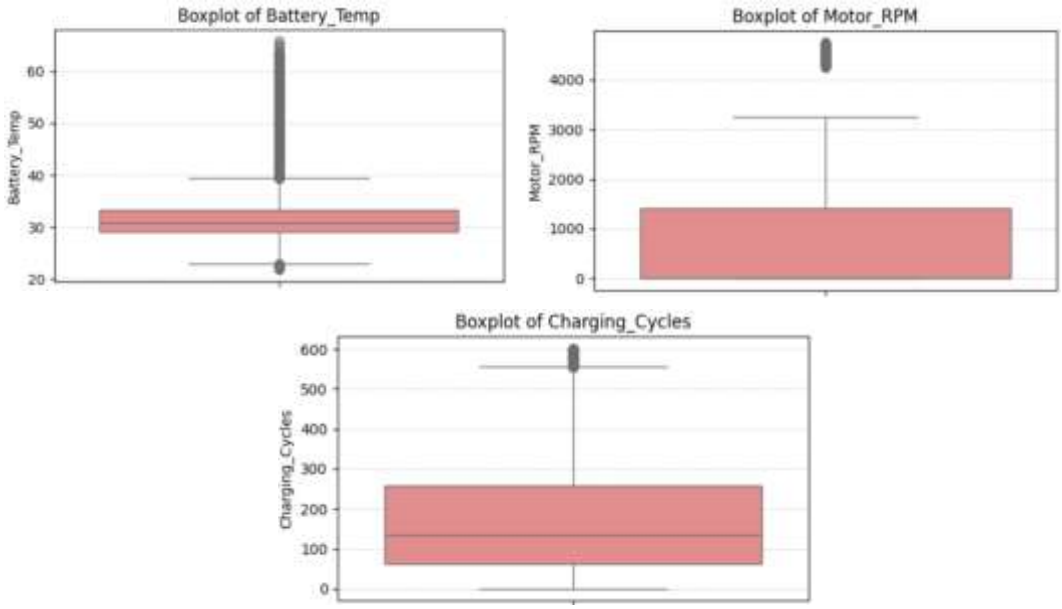

Figure 2: Outliers in the Numerical Features

In Fig. 2 Battery Temp shows that many values are over the upper whisker, it can means overheating cases. This information is important, because high battery temperaturs can reduced lifespan and it can bring some safety problems. The boxplot for Motor RPM shows many extreme values above 4000 RPM. These readings can be caused for sudden acceleration or aggressive driving patterns, especially from Heavy Users, those peaks also could indicate stress on the motor system and be important in failure predictions. Finally, Charging Cycles has many numbers of high outliers, with values up to 500 cycles. These is usually associated with long term and high usage vehicles. Due to frequent charging can damage battery health, it is important to consider this information for predicting battery failures. Also, other variables like Motor Temp and Motor Torque have moderate number of outliers. These variables were not removed because they can have useful information about vehicle stress and system warnings.

3.2.3 Correlation Analysis (Heatmap)

A correlation matrix was used to verified linear relationships between sensor variables. This helped to identify multicollinearity and redundancy in features.

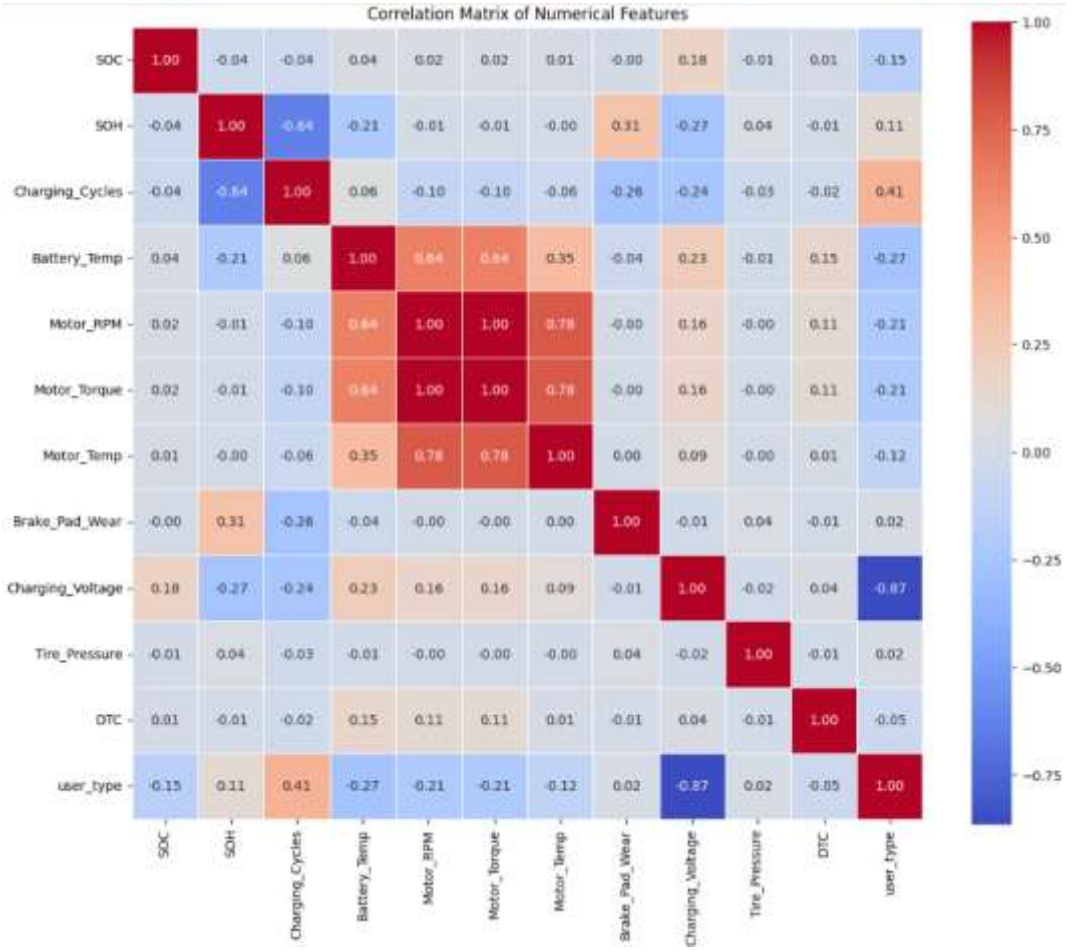

Figure 3: Correlation Matrix of Numerical Features

The Fig.3 shows that Motor RPM and Motor Torque correlation is perfect (1.00). This makes sense because higher RPMs means higher torque, especially in electric vehicles. Also, Motor Temp has a positive correlation with Motor RPM and Motor Torque (0.78), it suggests that an intense activity in the motor creates heat generation. This relationship is useful to anticipate failures for overheating. Finally, Charging Cycles and SOH show a negative correlation (-0.64), that indicates that frequent charging cycles can reduce the health of the battery.

3.2.4 Relationship between Features and Target (DTC)

T understand which variables can help to predict failures before they happen, we compared the distribution of numerical variables across the target variable DYC: Failure and No Failure. The boxplots shows important differences.

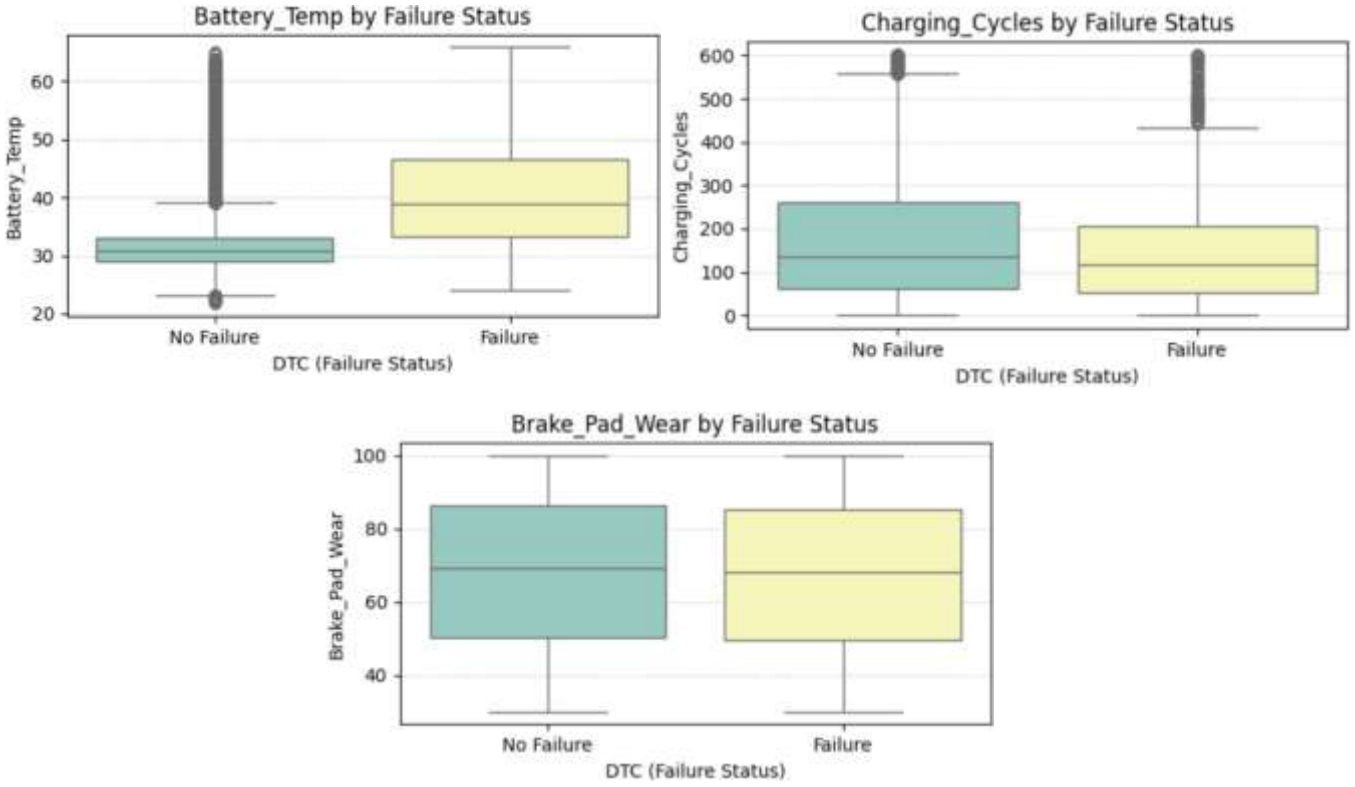

Figure 4: Relationship between main features with Target (DTC)

In Fig. 4 Battery Temp we can observe a clear change in distribution, where vehicles with failures have higher battery temperatures. This supports the idea that temperature stress can create problems in the system, it can be because of poor cooling or too much use in difficult driving conditions. On the other hand, Charging Cycles shows that contrary to initial assumptions vehicles without failures show higher charging cycles, this can suggest that vehicles that are frequently charging are better maintained or the usage makes to avoid the overcharging. Finally, we can see that Brake Pad Wear shows an increase in brake wear between the vehicles with failures, This could indicate that an aggressive driving or constant stop and go conditions generate stress on the system. Other variables like Motor RPM and Motor Torque show less differences, which makes them less important for predictions.

3.2.5 Comparative Analysis by User Type

To explore the behavior and mechanical diffrences between the user profiles, we compared distribution of important numerical variables. The following observations are based on the boxplots below.

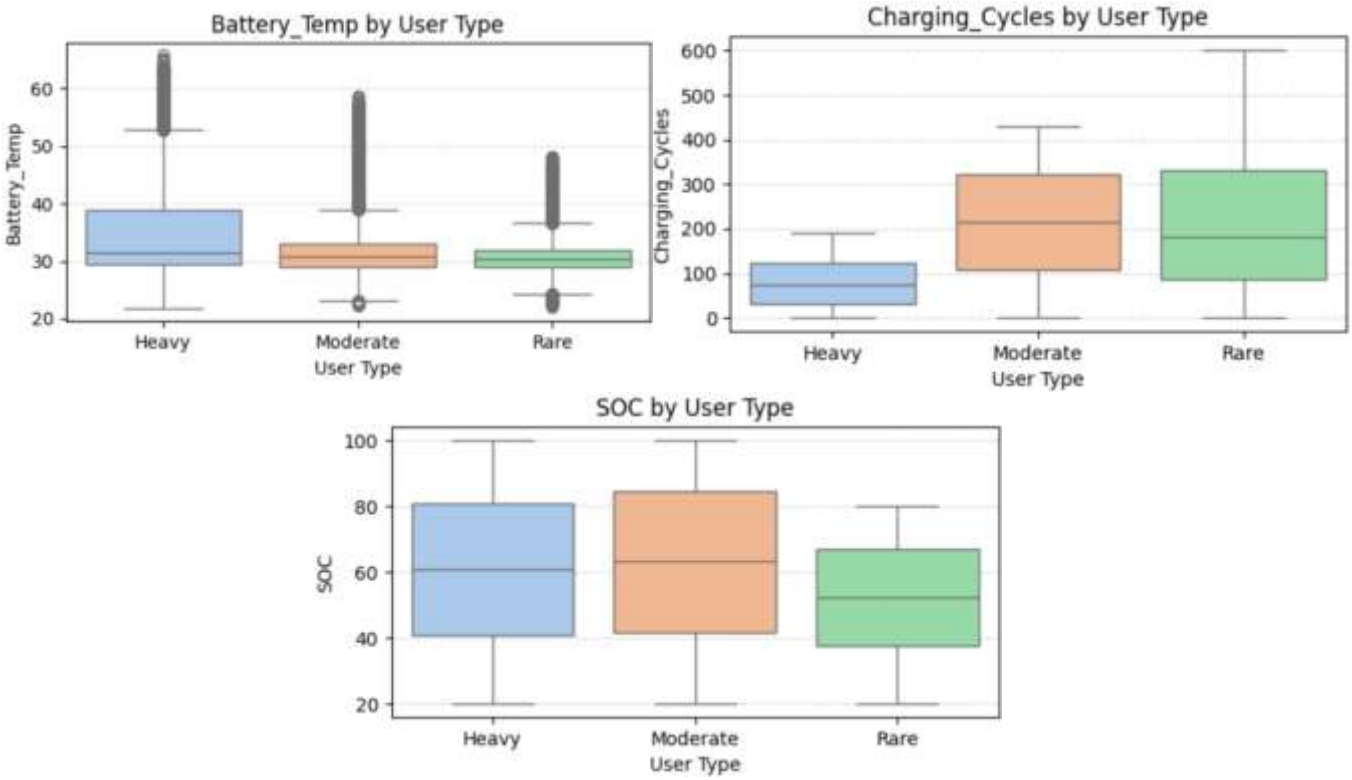

Figure 5: Distribution of Numerical Features by User Type

As Fig. 5 shows that for variables like Battery Temp, the heavy users often have higher temperatures with more extreme values. This suggests that they use more the vehicles and it can have thermal strain, which could increase wear. On the contrary, Rare users have the lowest battery temperatures because of the frequency of usage. In Charging Cycles case, Rare and Moderate users have higher values compared to Heavy users. This can be shorter but more frequent trips for Rare users or regular daily usage for Moderate users.

Heavy users may charge less frequently but they use the car more time and finally, SOC shows that Heavy users show the bigger variation, going from very low to totally charge. On the other hand, Rare users keep SOC lower, it can be possible because they have less driving activity and they charge less than the others.

3.3 Feature Engineering and Selection

Before training the models, it was necessary to prepare the dataset following some preprocessing steps. These steps included the transformation of the target variable, handling with missing or null values and the creation of one variable. The goal was to improve the performance of the model and make sure that the data that we use for training the model is consistent, representative and properly scaled when it was necessary.

3.3.1 Cleaning and Transformation of Target Column (DTC)

The original target variable, Diagnostic Trouble Codes (DTC) was not in a suitable format to work with binary classification. At the beginning this variable included different values to describe types of failures, so to simplify the work and achieve the objective, this variables was converted into two categories: Failure and No Failure. This step allowed us to focus to detect if the system is likely to fail or not, without considering the type of fail. This way to simplify is important in predictive maintenance, where the main goal is to alert about possible failures more than classify them.

3.3.2 Handling Missing Values

Before training the models, we checked the dataset trying to find any missing values. In this case, the dataset did not contain missing values in the main variables. This step confirmed that the dataset was ready to use it.

3.3.3 User Type Feature Creation

The original dataset had three different files, each one of them representing one vehicle usage profile: Rare**,** Moderate and Heavy User**.** To make a good analysis, we merged the three datasets in a single DataFrame.

For the next step we create a new column for User Type, this column shows the type of usage profile. This variable helped the model to study different pattern behaviors. However, we did not include the target variable in the training set because it can influence the probability of a DTC in different driving conditions.

3.3.4 Feature Selection Criteria

To ensure that the models were trained with useful variables, we worked only with numerical features. The selection process was based on a combination of exploratory data analysis (EDA), domain importance and statistical properties such as variance and correlation. Columns like identifiers or timestamps were removed, because they did not contribute to predict failures. The final set of variables captures different aspect of the vehicle performance, it includes temperature readings, torque levels, battery usage and brake conditions, among others.

3.3.5 Scaling Strategy (StandardScaler for Logistic Regression only)

Between the four models applied, only in the Logistic Regression model was necessary to use a Scaling strategy. This is because logistic regression is sensitive to the scale of input features, because it depends on gradient-based techniques for optimization. StandardScaler from scikit-learn was used to normalize the numerical variables and this transformation adjusted the features to have a mean of 0 and a standard deviation of 1, it improved the model performance and convergene. For the other models like Decision Tree, Random Forest and XGBoost was not necessary to apply scaling, because these models are based on tree structures and can make the decisions through threshold-based splits. It means that applying scaling to thee models would not have any effect on the performance or results.

3.4 Handling Class Imbalance

Many times imbalanced datasets can really affect the learning process and model performance. In this project, the target variable DTC showed a notable imbalance between the two classes, it means that most of the entries were part of the normal conditions (no failure) and only a smaller proportion were the failure cases. Since the main objective of this study is to predict failures before they happen, it is important to ensure that the model is not biased toward the majority class. To address this problem, we analyzed the original class distribution and then we applied oversampling techniques to compare the performance of the model before and after balancing the dataset.

3.4.1 DTC Class Distribution Before Balancing

Before to apply techniques to balance the data, the distribution of the target variable DTC was studied. The dataset showed a class imbalance, where most of the records were No failures = 0 and only a smaller portion was Failure = 1 class. This imbalance data can damage the performance of the model because it is going to favor the majority class during the

training model. The figure below shows the original distribution of both classes:

Figure 6: DTC class distribution before balancing.

3.4.2 SMOTE: Oversampling Strategy

To work on the class imbalance that we observed in the target variable DTC, we applied the Synthetic Minority Oversampling Technique (SMOTE). This is an important technique because it generates examples of the minority class and it helps to the model to learn better and reduces overfitting. The technique SMOTE was selected because of its ability to create more realistic samples improving the model sensitivity to minority class (failures) and maintain the general distribution of feature values in the dataset.

SMOTE was applied after splitting the data into training (105116, 10) and test (26280, 10) sets, but it was applied only to the training set. This ensures that the test data continues representative and do not lose important information during the model evaluation.

3.4.3 SMOTE Application Pipeline

The application of SMOTE followed a pipeline to ensure the data integrity and prevent loosing information. The process was implemented using the imblearn library within the scikit-learn framework. First, we divided the data in training and testing sets and we used Stratified split. Second, we applied SMOTE only to the training set, generating synthetic samples for the minority class. The technique was not applied to the test set to not affect the real data distribution. Then we trained the four models using the balanced training set, to make sure they learn from an equal representation of both classes. Finally, we did the evaluation in the original testing set, which still had the imbalance class. It can provide more real results of the performance of the model. This pipeline ensured that the created data did not affect the testing phase to compare the model performance before and after to apply SMOTE.

3.4.4 Distribution After Balancing

After we applied SMOTE to the training set, the distribution of the target variable DTC was balanced. The number of failure cases (DTC = 1) was increased through synthetic data to be equal with the majority class (DTC = 0) and have the same number of samples for both classes in the training data. This balancing was very important to mitigate the tendency of the

model to favor the majority class and to improve its ability identifying failure cases. This graph shows the new balanced distribution of the DTC variable in the training set after oversampling. Both classes now have the same number of instances.

Figure 7: DTC Class Distribution after Balancing

3.4.5 Rationale for Using SMOTE

The use of SMOTE (Synthetic Minority Oversampling Technique) was because we could identify a strong imbalance in the original dataset, the number of failure cases (DTC = 1) was lower than the no failure cases (DTC = 0), which represents a risk of bias during the model training. Without using the technique the models would likely perform well in predicting no failure cases but poorly in detecting real failures, which is the class we want to predict in the context of predictive maintenance.

SMOTE generates new samples of the minority class instead of only duplicate the ones that already exist. This helps to the model to learn better the patterns of failure cases while preserving diversity within the data. SMOTE was applied only to the training, it ensures that the evaluation on the test set continues without bias and more realistic. Balancing the classes, we can get more fair model training and improve the ability of the model to detect rare but important signals of failures.

4 Design Specification

4.1 Model Choice Rationale

To evaluate the performance of different approaches in predictive maintenance, we worked with four models: Logistic Regression, Decision Tree, Random Forest and XGBoost.

Table 1: Comparison of Selected Machine Learning Models for EV Fault Prediction

Model	Type	Feature Scaling Required	Handles Nonlinearity	Interpretability	Performance on Imbalanced Data	Reason for selection
Logistic Regression	Linear	Yes (StandardScaler)	No	High	Weak without balancing	Simple and interpretable baseline
Decision Tree	Tree-based	No	Yes	Medium	Good	Captures non-linear patterns
Random Forest	Ensemble of Trees	No	Yes	Low-Medium	Strong	Robust and generalizable
XGBoost	Gradient Boosted Trees	No	Yes	Low	Very strong	High performance in imbalanced datasets

Logistic Regression was selected like a baseline model, because it is very simple for interpretation. With this model we can evaluate if there is a lineal relationship between the variables and the target variable DTC. However, this model has some limitations whe it needs to work with complex data or no linear patterns. The Decision Tree model was included because they can study the relationships and interactions between variables without the need of scaling and even when they can overfit on small datasets, they usually offer transparent decision and also are easy to interpret.

Random Forest is an important model, it is able to reduce overfiting and improve generalization. It is also good working with noise data and can handle large datasets very well. However, it is not very easy to interpret but has a higher predictive accuracy. Finally, XGBoost was selected because its good performance as a gradient boosting algorithm. It is able to manage structured datasets and although sometimes it is not totally transparent, it has a good performance managing imbalanced data. This combination of models is important to balance performance and interpretability. The table below shows the main characteristics of each model:

4.2 Model Pipeline

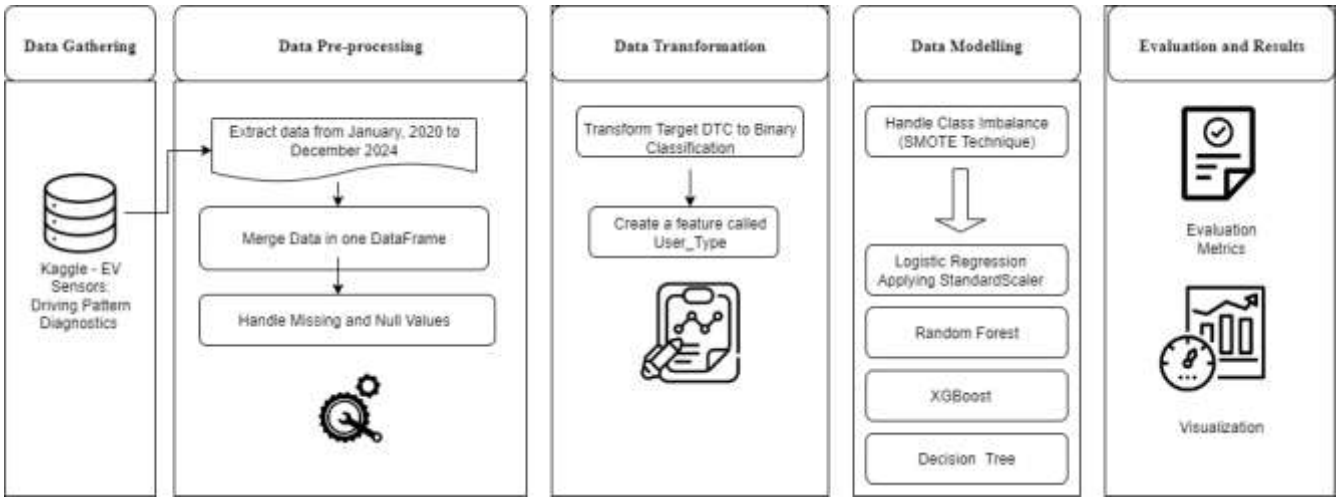

Figure 8: Pipeline of the Model

After the preparation data that include cleaning, feature selection and transformation of the target variable, the process of the model follow a structured pipeline. First, the dataset was divided in features (x) and the target variable (y) and columns that do not have important information like timestamp or IDs were not considered. Before training the model, the dataset was split in training and testing sets, for this we use a standard ratio 80/20. This heped to reduce bias and overfiting. Also, to balance the data, SMOTE was applied only to the training set to ensure that the oversampling do not affect the test data.

For Logistic Regression we used StandardScaler, this model is sensitive to the magnitude of features but Decision Tree, Random Forest and XGBoost do not need to be scaled because of its structure. Each model was trained independently, using Scikit-learn and XGBoost libraries. The predictions on the test set were evaluate using evaluation metrics like: Accuracy, Balanced Accuracy, F1 Score, Precision, Recall and ROC – AUC. All the pipeline was implemented in Python ad we use Jupyter Notebooks and machine learning libraries like Pandas, Scikit-Learn and Imblearn.

4.3 Rationale for Metric Selection

In this research, we used different evaluation metric to measure the prformancec of the model. Due to the target variable DTC is binary and e worked with an imbalanced dataset it was necessary to use more metrics than accuracy, these metrics were added to understand better the advantages and disadvantages of the model.

We used Accuracy to get the overall percentage of the correct predictions but in imbalanced datasets, a high accuracy can predict more the majority class and we do not get the correct results. Precision shows how many of the predicted failures were correct and it helps to avoid the false alarms. We use Recall to know hoy many failures were correctly detected. F1 Score is a combination between precision and recall into a single value, this metric is useful when we need a balance to avoid false positives and get real failures. Balanced Accuracy helps to adjust the accuracy in class imbalance, it gives the same importance to both classes and also helps to evaluate how well the model can handle rare failure cases. Finally, we ue ROC AUC to measure the capacity of the model to separate both classes in different scenarios, a higher ROC AUC . A combination of all these metrics was necessary to evaluate the model and be able to select the most appropriate for predictive maintenance.

5 Implementation

5.1 Overview of the Implementation Environment

To implement this project we use Python, all the experiments were made using Jupyter Notebooks because they provide a good interface to develop and test machine learning models. The most important libraries that we used in this implementation were: Pandas and Numpy to handle the data and make numerical operations, Matplotlib and Seaborn to create visualizations, Scikit-learn for implementing machine learning model and metrics, Imbalanced-learn (imblearn) to handle the imbalance data using SMOTE and XGBoost to implement the XGBoost classifier.

5.2 Data Loading and Preprocessing

The dataset that we used in this project was created merging three CSV files that were obtained from Kaggle, each file represent different vehicle user types: heavy, moderate and rare. After loading the files to Jupyter all the columns that were not necessary for the analisis were removed, also we create a new column called User_Type and finally the target variable DTC was converted to binary, where 0 means no failure and 1 means failure.

5.3 Model Training Process

To predict the target variable DTC we use four machine learning models using the same training and test sets in all the models to be able to do a good and fair comparison. The models that we used were:

- **Logistic Regression (LR)**
- **Decision Tree (DT)**
- **Random Forest (RF)**
- **XGBoost (XGB)**

We trained each model using the imbalanced dataset and we applied Smote to balance the data. However, the test set did not change to reflect the original class distribution and true results of the performance. For Logistic Regression, we applied StandardScaler from scikit-learn. The other models, which are tree-based, did not require scaling.

For training the model we use hyperparameters in all the models. Also we used the .fit() method to train each model using the training set and .predict() to make predictions on the test set and the same evaluation process was applied to all the models.

5.4 Implementation Challenges

During the development of this project there were some challenges to work, for example we had to manage the imbalanced dataset because it had a low number of failure cases DTC = 1, it affected the performance of the model, especially in recall. To manage this problem we use SMOTE, it helped to improve the detection of failures. Another problem was that all experiments were run on a personal laptop. While in this case it was enough to run the models, there was less opportunities to run more complex experiments or apply another kind of techniques. However, at the end we implemented the pipeline and had useful results.

6 Evaluation

In this chapter we describe the results from the development of the machine learning models. The evaluation focuses on the comparison of the model performance before and after applying SMOTE, it means that we compared the results with imbalance and balance data. Different evaluation metrics were used to measure how effective was each model. The analysis highlights the impact of the imbalance data on the model performance and helps to identify the most suitable approach to detect failures. To understand better the results we included some visualizations, tables and confusion matrices.

6.1 Model Performance without SMOTE

6.1.1 Performance Metrics on Imbalanced Data

In this case, this experiment evaluates the performance of four models on the original dataset, which was imbalance, where the failure cases were not properly represented. The main objective was to see how this imbalance can affect the ability of the model to detect failures.

Table 2: Evaluation Metrics on Imbalanced Data

Model	Accuracy	Balanced Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	0.9823	0.5000	0.0000	0.0000	0.000	0.8369
Decision Tree	1.0000	1.0000	1.0000	1.0000	1.000	1.0000
Random Forest	1.0000	1.0000	1.0000	1.0000	1.000	1.0000
XGBoost	0.9955	0.9164	0.9023	0.8344	0.867	0.9971

As we can see in table 2, the Logistic Regression model did not have a good performance detecting failure cases, even when it had a high accuracy, the recall and precision were zero, which means that all the predictions came from the majority class. This is a common problem when we train models without using any technique to correct imbalance datasets. On the contrary, the other models performed very well, Decision Tree and Random Forest got perfect scores in all the metrics and XGBoost also got high results. However, such perfect performance can suggest an overfiting. These results confirm that imbalance data can have an important impact on simple models.

6.1.2 Confusion Matrix Analysis on Imbalanced Data

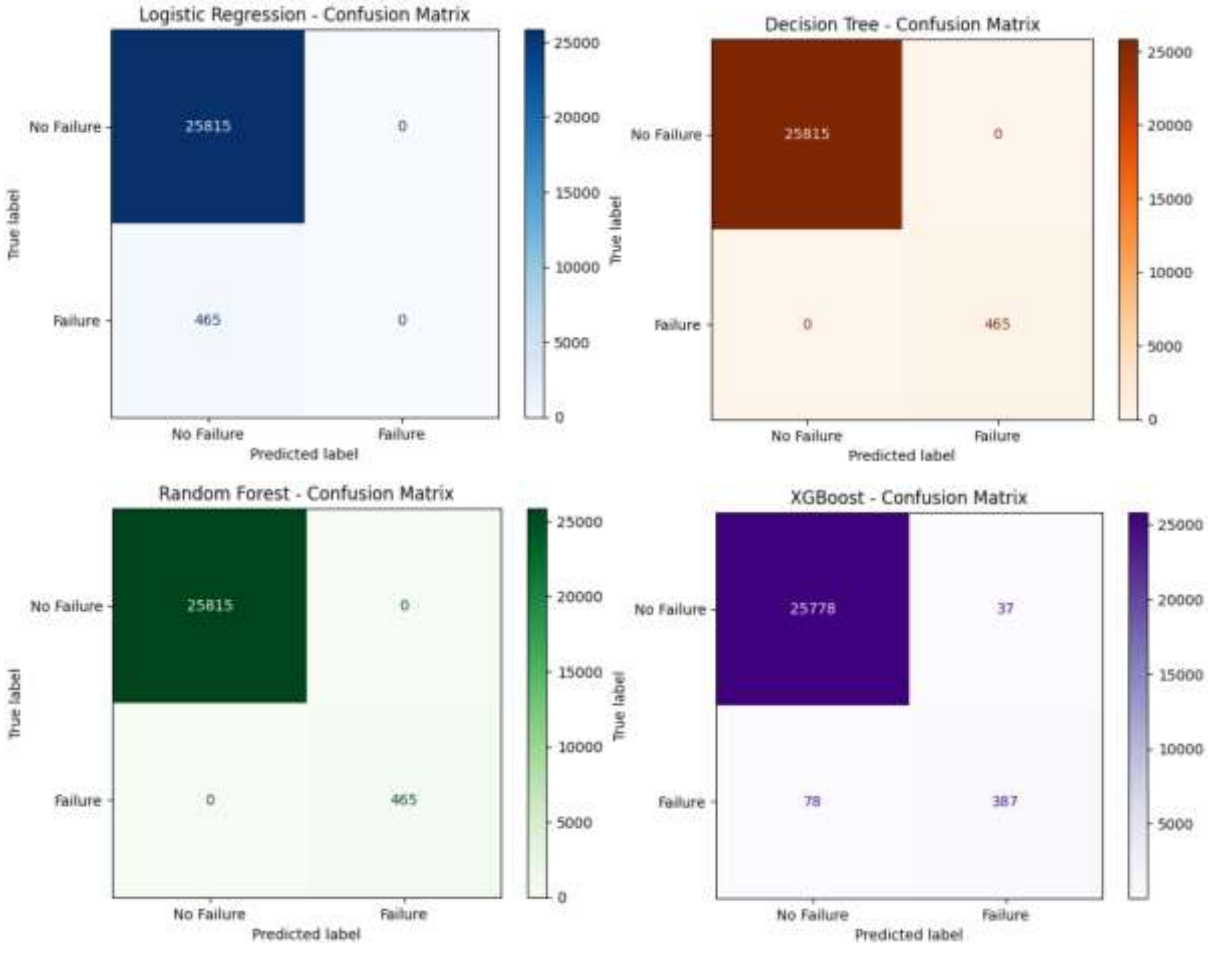

Figure 9: Confusion Matrix on Imbalanced Data

A confusion matrix presented in Fig. 9 demonstrates that the existing model is responding to the data that is currently imbalanced. Logistic Regression was labelling all samples as the No Failures and was skipping the actual failures. It is possible because the model was focused on the majority class. In contrast, Decision Tree and Random Forest did not have errors. Despite these results looking perfect, these are not a reflection of the true behavior and can be a sign that the model overffits the traininng data. Finally, XGBoost made a few mistakes but was very accurate because it was able to detect a few extra cases of failure and fewer false alarms. In this case, for generalization if we get few errors can be better than perfect scores.

6.2 Model Performance with SMOTE

6.2.1 Performance Metrics on Balanced Data

Table 3: Evaluation Metrics on Balanced Data

Model	Accuracy	Balanced Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression	0.7508	0.8087	0.0586	0.8688	0.1098	0.8296
Decision Tree	0.9909	0.9932	0.6605	0.9957	0.7942	0.9932
Random Forest	0.9990	0.9974	0.9527	0.9957	0.9737	0.9999
XGBoost	0.9969	0.9794	0.8748	0.9613	0.9160	0.9946

After balancing the data using SMOTE, the performance of the models improved. Logistic Regression could detect most of failures with a recall of 87%. However, precision was 5.8% , which means that the model got many false alarms and it was very sensitive to the minority class. This shows that even when the model tried to get failures it was not good for separating them from normal cases. Decision Tree got good results, with a high precision, recall and F1 Score, it means that it was able to detect almost all the failures without any mistakes. Random Forest performed better with a precision almost 100% and also a high F1 Score, it implies that this model dealt well with the balancing data and it provided better predictions. As well, XGBoost performed well since it identified over 96% of actual failures and it presented a better precision value than the other models and maintaining a good balance between identifying actual failures and avoiding false ones. Therefore, Random Forest and XGBoost presented a more consistent performance and Logistic Regression was not very effective for actual cases.

6.2.2 Confusion Matrix Analysis on Balanced Data

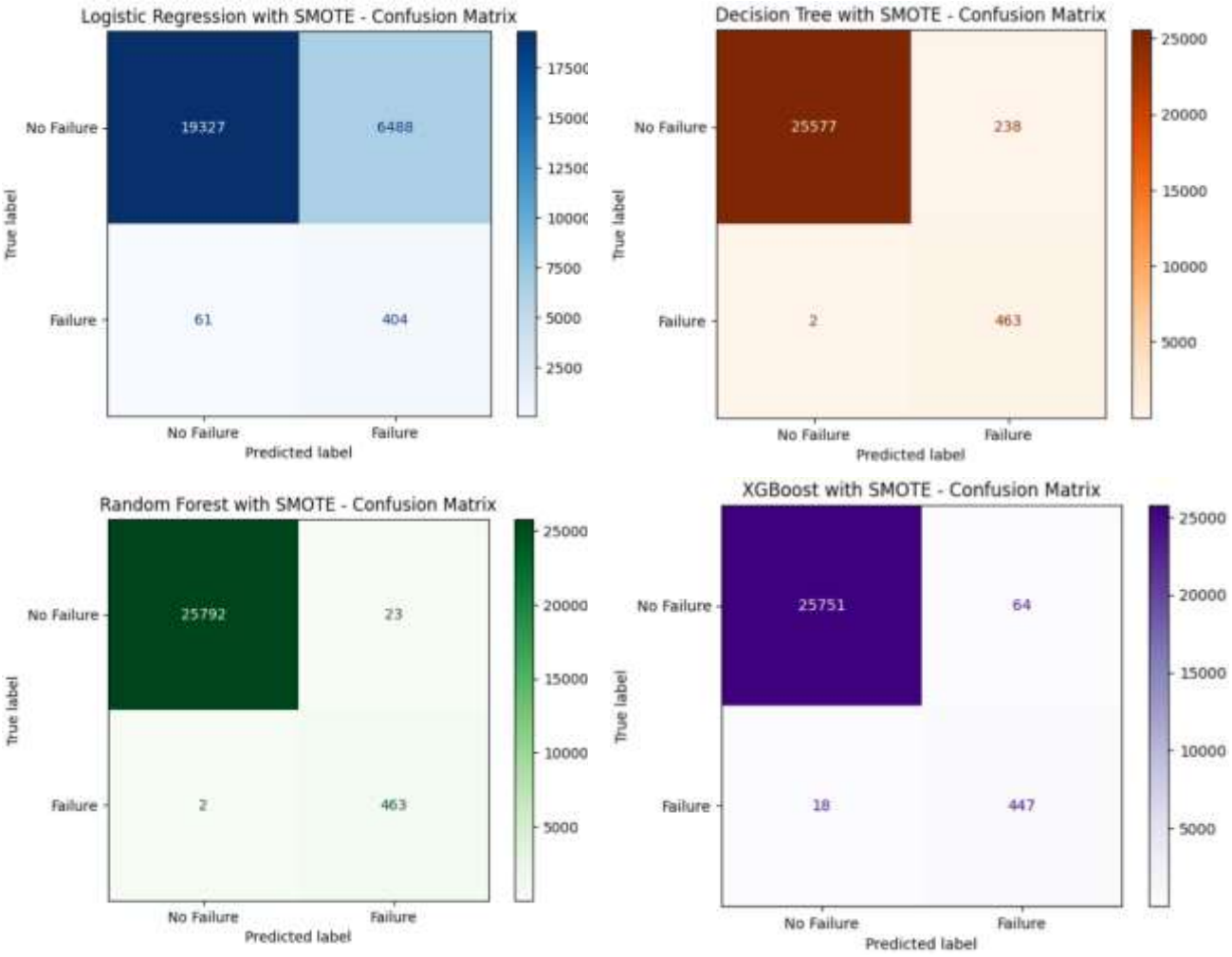

Figure 10: Confusion Matrix on Balanced Data

Confusion matrix Fig. 10 implies the results from models that were trained on balanced data, as we can notice, even though Logistic Regression had a good recall for failures with 404 out of 465 correctly identified, it also had a high rate of false positives. Decision Tree presented better results with only 2 false negativves and 238 positives, it means that it recognizes on both classes. Nonetheless, Random Forest performed very well, it only had 2 false negativs and 23 false positives, the graph shows a difference between the two classes and it demonstrates that this particular model is good to identify failures. Furthermore, CGBoost achieved good results with 18 false negatives and 64 false positives, even though it presented more false negatives and false positives than Random Forest. In conclusion, balancing the data using SMOTE improved the performance of all the models

Comparison and Analysis of Results

6.2.3 Comparison Table – Imbalanced vs Balanced Models

This table provides a comparison of the four machine learning models that were used to predict the target variable Diagnostic Trouble Codes DTC, these results shows both experiments. The main objective is to see the differences between using balance and imbalance data.

Table 4: Comparison Evaluation Metrics Results

Model	Accuracy	Balanced Accuracy	Precision	Recall	F1 Score	ROC AUC
Logistic Regression (Imb)	0.9823	0.5000	0.0000	0.0000	0.0000	0.8369
Logistic Regression (Bal)	0.7508	0.8087	0.0586	0.8688	0.1098	0.8296
Decision Tree (Imb)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Decision Tree (Bal)	0.9909	0.9932	0.6605	0.9957	0.7942	0.9932
Random Forest (Imb)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Random Forest (Bal)	0.9990	0.9974	0.9527	0.9957	0.9737	0.9999
XGBoost (Imb)	0.9955	0.9164	0.9023	0.8344	0.8670	0.9971
XGBoost (Bal)	0.9969	0.9794	0.8748	0.9613	0.9160	0.9946

In this case, we can observe that Logistic Regression shows a bad performance when working with imbalance data, where F1 Score = 0.0, Precision is still low but it improves in Recall after SMOTE from 0.00 to 0.86. On the other hand, Decision Tree got perfect scores when working without SMOTE, it may indicate overfiting, however whit balance data the results dropped but become more realistic F1 = 0.79. Finally, Random Forest and XGBoost had a good performance working with both datasets but they improved with SMOTE.

6.2.4 Performance Comparison by Metric per Model

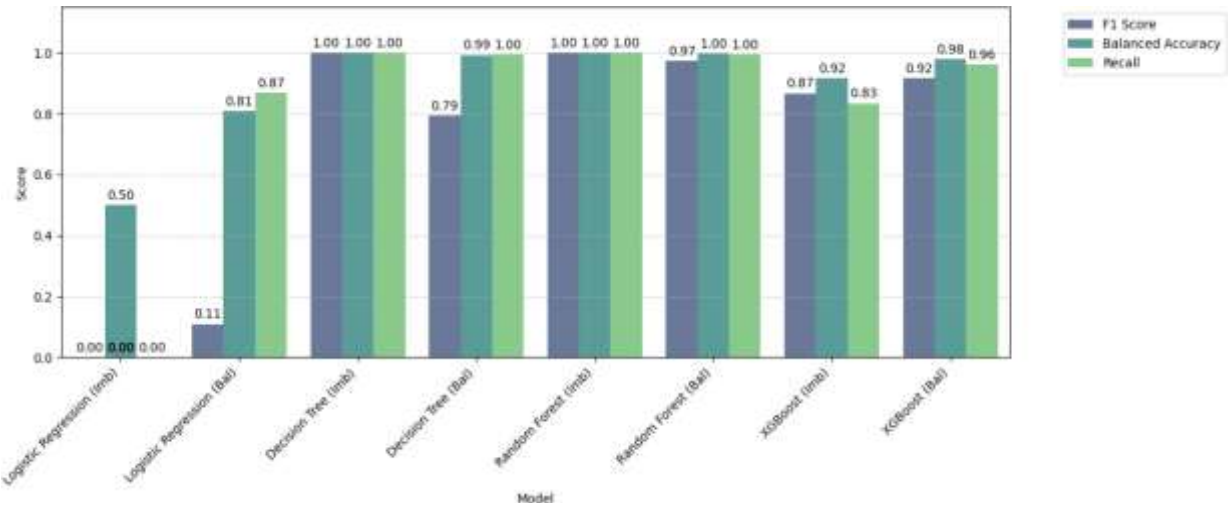

Figure 11: Evaluation Metrics by Model

The Graph Fig. 11 shows a comparison of the evaluation metric by model. Logistic Regression had a poor performance when working without SMOTE because it missed almost all the positive cases, on the contrary when we applied SMOTE it got a higher Recall but the performance in general was still low. Other models like Decision Tree, Random Forest and XGBoost had very high scores whe working without SMOTE, it may be becaue they were more focused on the majority class, however after using SMOTE their results continue strong but they appear more realistic. In general, Random Forest and XGBoost gave the best and most stable results.

6.2.5 ROC AUC Comparison Chart

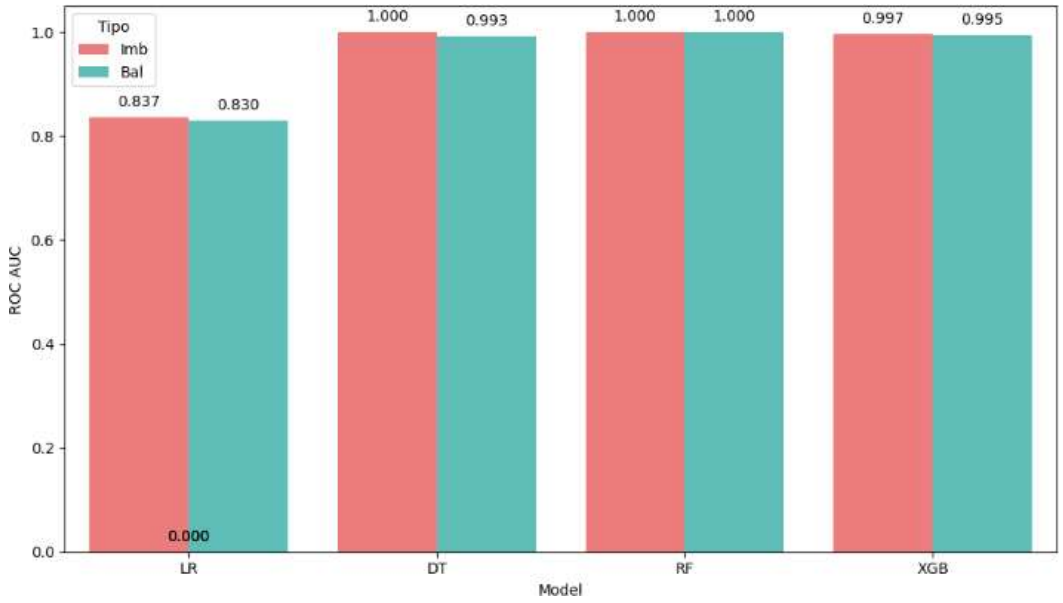

Figure 12: ROC AUC Comparison Chart

This Graph Fig 12. Compares the performance of the models to find the difference between failures a no failures using ROC AUC. All the models that were trained without SMOTE show high values, especially Decision Tree and Random Forest 1.00, it could suggest that they only memorized the data and did not learn general patterns. On the other hand, when we applied SMOTE the results were still high but the differences showed that they looked more realistic, especially Decision Tree and Logistic Regression. In general, the performance of all the models was very good in terms of ROC AUC.

6.3 Discussion

The final results showed that models like Random Forest and XGBoost perform very well to predict failures, even before to balancing the data. These models performed well with high precision and recall, which shows its ability to work with imbalanced data (Attia and Aoulmi. 2025). After applying SMOTE, theirs were still high and appeared balanced, which reflects that they are suitable for predictive maintenance it confirms the value of handling class imbalance described by Mahale et al. (2025).

Logistic Regression, on the other hand, did not work well when the data was imbalanced, it could not identify failures as we could see in other studies, this kind of model is not good handling imbalanced data (Buabeng et al. 2021) and when SMOTE was used on it, improved on its recall but provided too many false positives. Therefore, simple models might not work for this type of study unless the data is tuned specifically for them.

As a general overview, performance of the model was good and the results were consistent with what were expected. The use of SMOTE was beneficial in most scenarios and the experiments served to understand which models are appropriate for this type of prediction. The goal of experimenting different models and checking their behavior was achieved.

7 Conclusion and Future Work

This research wanted to answer three main questions: how accurate machine learning models can predict Diagnostic Trouble Codes (DTC) in electric vehicles, which sensor variables have more influence on predictions and if balancing the dataset with SMOTE improves the performance of the model. The objective was to evaluate and compare four supervised models: Logistic Regression, Decision Tree, Random Forest and XGBoost using real EV sensor data.

The study was able to answer these questions. Random Forest and XGBoost provided the best performance with a good balance between precision and recall when we used SMOTE. Logistic Regression can increase recall when is dealing with balanced data but showed many false positives. Battery temperature, brake pad wear and charging cycles were the most important variables, which is consistent with the literature review that we did on predictive maintenance. The models worked well, but the work has various limitations since the data was based on history and not on current data from EVs.

In future work it is possible to expand this study by moving to multi-class prediction to classify different types of failures, applying deep learning methods such as LSTMs to capture temporal patterns and testing models on real-time fleet data to improve robustness in operational conditions. Also, tuning of model hyperparameters and exploring hybrid systems that integrate predictive models with maintenance scheduling tools could improve both accuracy and practical usability.

References

Achariyee, B., Ghosh, R.R., Das, S., Nandan, D. and Dey, T. (2021). *A novel architecture for predictive maintenance and real-time monitoring of electric vehicle*. In: *Proceedings of the 2021 3rd International Conference on Intelligent Engineering and Management (ICIEM)*, pp. 213–217. IEEE. https://doi.org/10.1109/TENCON61640.2024.10902745

Amihai, I., Pareschi, D., Gitzel, R., Subbiah, S., Sosaie, G. and Kotriwala, A.M. (2018). *An industrial case study using vibration data and machine learning to predict asset health*. In: *2018 IEEE 20th Conference on Business Informatics (CBI)*, Vol. 1, pp. 125–134. IEEE. https://doi.org/10.1109/CBI.2018.00028

Assagaf, I., Subekti, M.V., Putra, A.R. and Susanto, M.T. (2023). *Machine predictive maintenance by using support vector machines*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/368365294_Machine_Predictive_Maintenance_by_Using_Support_Vector_Machines [Accessed 12 Sep. 2025].

Attia, M. and Aoulmi, Z. (2025). *Improving electric vehicle maintenance by advanced prediction of failure modes using machine learning classifications*. *Eksploatacja i Niezawodnosc – Maintenance and Reliability*, 27(3), pp. 179–187. https://doi.org/10.17531/ein/201372

Aydın, C. and Eventuğ, B. (2025). *Evaluation of predictive maintenance efficiency with the comparison of machine learning models in machining production process in brake industry*. *PeerJ Computer Science*, (July 2025), p. e2999. https://doi.org/10.7717/peerj-cs.2999

Baheti, M., Shankarnarayan, V.K., Dhanirajan, S., Chandak, S., Gurunathan, T. and Chandrasekar, P. (2024). *AI-powered predictive maintenance for electric vehicle fleets*. In: *Proceedings of the 2024 Asian Conference on Intelligent Technologies (ACOIT)*, pp. 6–7. IEEE. https://doi.org/10.1109/ACOIT62457.2024.10939072

Benhanifia, A., Ben Cheikh, Z., Moura Oliveira, P., Valente, A. and Lima, J. (2025). *Systematic review of predictive maintenance practices in the manufacturing sector*. *Intelligent Systems with Applications*, 26, p.200501. https://doi.org/10.1016/j.iswa.2025.200501

Buabeng, A., Simons, A., Frempong, N.K. and Ziggah, Y.Y. (2021). *A novel hybrid predictive maintenance model based on clustering, SMOTE and multi-layer perceptron neural network optimised with grey wolf algorithm*. *SN Applied Sciences*, 3, p.504. https://doi.org/10.1007/s42452-021-04598-1

Chukwudi, I.J., Zaman, N., Rahim, M.A., Rahman, M.A., Alenezi, M.J.F. and Pillai, P. (2024). *An ensemble deep learning model for vehicular engine health prediction*. *IEEE Access*, Advance online publication. https://doi.org/10.1109/ACCESS.2024.3395927

Jafari, S., Yang, J.-H. and Byun, Y.-C. (2024). *Optimized XGBoost modeling for accurate battery capacity degradation prediction*. *Results in Engineering*, 24, p.102786. https://doi.org/10.1016/j.rineng.2024.102786

Jauhar, S.K., Sethi, S., Kamble, S.S., Mathew, S. and Belhadi, A. (2024). *Artificial intelligence and machine learning-based decision support system for forecasting electric vehicles' power requirement*. *Technological Forecasting and Social Change*, 204, p.123396. https://doi.org/10.1016/j.techfore.2024.123396

Kumar, A. (2023). *Machine learning algorithms for predictive maintenance in industrial environments: A comparative study*. *Journal of Artificial Intelligence General Science (JAIGS)*, 2(1), pp.1–12. [online] Available at: https://www.researchgate.net/publication/380454074_Machine_Learning_Algorithms_for_Predictive_Maintenance_in_Industrial_Environments_A_Comparative_Study [Accessed 12 Sep. 2025].

Kumar, R.S., Singh, A.R., Narayana, P.L., Chandrika, V.S., Bajaj, M. and Zaitsev, I. (2025). *Hybrid machine learning framework for predictive maintenance and anomaly detection in lithium-ion batteries using enhanced random forest*. *Scientific Reports*, 15, p.90810. https://doi.org/10.1038/s41598-025-90810-w

Lou, J. (n.d.). *Comparative analysis of logistic regression, random forest, and XGBoost for click-through rate prediction in digital advertising* [Manuscript]. Nanjing University of Information Science and Technology. https://doi.org/10.2991/978-94-6463-542-3_54

Mahale, Y., Kolhar, S. and More, A.S. (2025). *Enhancing predictive maintenance in automotive industry: Addressing class imbalance using advanced machine learning techniques*. *Discover Applied Sciences*, 5, p.66. https://doi.org/10.1007/s42452-025-06827-3

Mehra, K. (2025). *EV sensors: Driving pattern diagnostics (2020–2024)*. [dataset] Kaggle. Available at: https://www.kaggle.com/datasets/kunalm95/ev-sensors-driving-pattern-diagnostics-2020-24 [Accessed 12 Sep. 2025].

Mishra, P.S., Makhdoomi, S. and Sinha, A. (2024). *Fault detection and diagnosis of electric vehicles using artificial intelligence: A review*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/383631967_Fault_detection_and_diagnosis_of_electric_vehicles_using_artificial_intelligence [Accessed 12 Sep. 2025].

Nunes, P., Santos, J. and Rocha, E. (2023). *Challenges in predictive maintenance – A review*. *CIRP Journal of Manufacturing Science and Technology*, 40, pp.53–67. https://doi.org/10.1016/j.cirpj.2022.11.004

Prytz, R. (2014). *Machine learning methods for vehicle predictive maintenance using off-board and on-board data* (Master's thesis). Halmstad University, Sweden. [online] Available at: https://www.researchgate.net/publication/273452541_Machine_learning_methods_for_vehicle_predictive_maintenance_using_off-board_and_on-board_data [Accessed 12 Sep. 2025].

Spiegel, S., Mueller, F., Wiesmann, D. and Bird, J. (2018). *Cost-sensitive learning for predictive maintenance*. *arXiv*. https://doi.org/10.48550/arXiv.1809.10979

Surabhi, S.N.R.D. (2024). *Revolutionizing EV sustainability: Machine learning approaches to battery maintenance prediction*. *Educational Administration: Theory and Practice*, 29(2), pp.355–376. https://doi.org/10.53555/kuey.v29i2.4230

Theissler, A., Pérez-Velázquez, J., Kettelgerdes, M. and Elger, G. (2021). *Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry*. *Reliability Engineering & System Safety*, 215, p.107864. https://doi.org/10.1016/j.ress.2021.107864

Tsylenko, M.D., Gaidel, A.V. and Kupiyanov, A.V. (2022). *Comparison of feature selection algorithms for data classification problems*. In: *Proceedings of the 2022 International Conference on Industrial Engineering*, 5(1), pp.20–25. IEEE. https://doi.org/10.1109/ITNT55410.2022.9848765

Zhang, X., Zou, M., Hu, S. and Sun, Z. (2021). *Prognostics and health management design for reliability prediction of electric vehicle battery using machine learning*. *Reliability Engineering & System Safety*, 215, p.107851. https://doi.org/10.1016/j.ress.2021.108063