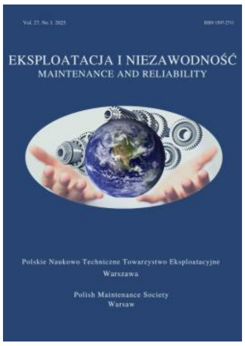


Eksploatacja i Niezawodnosc – Maintenance and Reliability
Volume 27 (2025), Issue 3
journal homepage: http://www.ein.org.pl

Article citation info:
Attia M, Aoulmi Z, Improving Electric Vehicle Maintenance by Advanced Prediction of Failure Modes Using Machine Learning Classifications, Eksploatacja i Niezawodnosc – Maintenance and Reliability 2025: 27(3) http://doi.org/10.17531/ein/201372

Improving Electric Vehicle Maintenance by Advanced Prediction of Failure Modes Using Machine Learning Classifications

Indexed by: Web of Science Group

Moussa Attia[a,*], Zoubir Aoulmi[a]

[a] Environment Laboratory, Institute of Mines, Echahid Cheikh Larbi Tebessi University, Tebessa 12002, Algeria

Highlights

- Machine learning enhances predictive maintenance for electric vehicles.
- Advanced algorithms identify potential failure modes before they occur.
- Improved efficiency and reduced downtime through accurate failure predictions.
- Optimised maintenance schedules based on real-time vehicle performance data.
- Data-driven insights extend the lifespan of electric vehicle components.

Abstract

This study stands out for its novelty, offering an alternative solution to traditional methods for analyzing failure modes and their effects. We utilized machine learning techniques, which have enabled a significant shift in the predictive maintenance of electric vehicles. We performed numerous tests and evaluations of advanced models such as random forests, decision trees, logistic regression, and neural networks, where random forests and neural networks achieved exceptional accuracy of 96.67%. This breakthrough improves fault prediction accuracy, reduces operational costs, and minimizes downtime by combining numerical and categorical data. The study focuses on the transformative potential of machine learning, enhancing the reliability, lifespan, and maintenance of electric vehicles through a data-driven approach. The main innovation of this study lies in integrating multiple models, such as Random Forest and Neural Networks, to analyze failures in electric vehicles. While previous studies typically relied on traditional techniques like decision trees or regression analysis, our research presents a multi-layered approach, enabling the models to detect more complex patterns and improve prediction accuracy. Moreover, we incorporate real-world data collected from electric vehicle sensors, which allows the model to make precise predictions in real-world operational environments. This approach significantly advances previous studies, which primarily relied on simulated data or isolated models.

Keywords
predictive maintenance, machine learning, electric vehicles, random forests, neural networks, advanced data

1. Introduction

The rapid transition towards electric vehicles (EVs) presents new challenges in maintaining the reliability and performance of these vehicles (1). Unlike traditional internal combustion engine (ICE) vehicles, EVs rely on complex electronic components, such as high-voltage batteries, electric motors, and associated control systems, introducing new failure modes that traditional maintenance methods cannot fully address (2-4). These failures can significantly impact operational efficiency, leading to unplanned downtime and higher maintenance costs (5, 6). To mitigate these issues, predictive maintenance using machine learning (ML) techniques has become a pivotal strategy in the automotive industry, including for EVs, to

(*) Corresponding author.
E-mail addresses: M. Attia (ORCID: 0000-0003-3700-7049) moussa.attia@univ-tebessa.dz, Z. Aoulmi (ORCID: 0000-0001-7212-7332) zoubir.aoulmi@univ-tebessa.dz

enhance reliability and minimize downtime (7).

Integrating ML models such as Random Forest (RF) and Neural Networks (NN) has shown significant promise in predicting failures in EV systems (8-10). These models can detect complex failure patterns by analyzing sensor data from various EV components (11, 12). Recent studies have demonstrated the effectiveness of these models, particularly in predicting failures in battery and motor control systems. This section synthesizes findings from multiple studies, highlighting the current state and challenges of ML-based predictive maintenance in EVs (10, 13, 14).

1.1. Effectiveness of Machine Learning Models:

- Random Forest and Neural Networks:

Sheikh et al. (2024) highlighted the superior performance of RF models in predicting battery failures, achieving over 90% accuracy, which surpasses traditional methods such as Failure Mode and Effects Analysis (FMEA) (15). Similarly, Neural Networks have been employed to capture complex patterns in sensor data, improving predictive accuracy by handling non-linear relationships and interactions between variables that traditional methods struggle to detect.

- Long Short-Term Memory (LSTM) Networks:

Li et al. (2023) demonstrated using LSTM networks to predict battery degradation. LSTM networks are particularly effective for capturing temporal dependencies in sensor data, making them more suitable for time-series analysis than simpler models that cannot account for these patterns (15).

1.2. Hybrid Models and Their Advantages:

- Combination of ML Techniques:

Hybrid models, which integrate different ML techniques, are gaining traction due to their ability to improve prediction accuracy. Ullah et al. (2022) proposed a hybrid approach combining Random Forest (RF) and Support Vector Machines (SVM) to predict failures in motor control systems. This combination improved accuracy and operational efficiency by leveraging the strengths of both models in handling different aspects of the data (15).

- Integration with Deep Learning:

Peres et al. (2023) explored integrating deep learning methods with traditional ML models, such as RF and SVM, to handle large and complex datasets. This integration enhances the robustness of the prediction models, making them more capable of managing high-dimensional data and improving prediction reliability (15).

1.3. Challenges and Limitations:

- Data Quality and Real-World Applicability:

One of the significant challenges in predictive maintenance is the reliance on simulated datasets or isolated ML models, which limits their applicability in real-world scenarios. The complexity and high dimensionality of real-world EV data, particularly when integrating data from multiple sensors, present significant challenges (16). Studies such as Ghelani (2024) emphasize the importance of high-quality, real-world data for training reliable models that can function effectively in operational environments.

- Computational Complexity:

As noted by Lorenti et al. (2023), deep learning models require significant computational resources, which can limit their feasibility for real-time applications in resource-constrained environments. The computational demands of deep learning models can hinder their deployment, especially in real-time predictive maintenance scenarios where speed and efficiency are critical (17).

1.4. Broader Perspectives and Future Directions:

While the advancements in ML-based predictive maintenance for EVs are promising, several broader perspectives should be considered. Integrating domain knowledge and human expertise into ML algorithms can enhance the predictive accuracy and relevance of the models. Ghelani (2023) discusses how incorporating expert insights can help refine predictions, especially in complex scenarios where data alone might not provide a full understanding of failure mechanisms (16).

Additionally, the role of edge computing and distributed ML techniques is gaining attention in enabling real-time predictive maintenance, particularly in remote or resource-constrained environments (16). These approaches allow for the deployment of models on-site, reducing the need for constant cloud connectivity and enabling quicker decision-making processes. As Ghelani (2024) explored, this is an ongoing research area.

Finally, ethical considerations like privacy protection and regulatory compliance are crucial when deploying ML-based predictive maintenance solutions. Researchers, including

Ghelani (2024), emphasize the importance of addressing privacy concerns related to the data used in these systems, mainly when dealing with personal or sensitive data from vehicle sensors.

This study's innovative approach of integrating multiple machine learning models, specifically Random Forest and Neural Networks, using real-world data significantly enhances predictive accuracy. It provides deeper insights than traditional methods reliant on isolated models or simulated data.

2. Methodology

2.1. Data Collection and Preprocessing

A dataset of real-world data collected from multiple sensors in electric vehicles, including high-voltage batteries, electric motors, and control systems, was used. The data provides insights into electric vehicles' thermal performance, energy consumption, and charge status. This data was gathered from several electric cars operating in various environments, encompassing over 300 data points ranging from minor faults to complete failures. The dataset is representative of the real-world operational conditions of electric vehicles.

Data preprocessing was crucial in cleaning and organizing the dataset for subsequent analysis. The primary tasks involved addressing missing values, encoding categorical variables, and normalizing numerical features to ensure consistency across the dataset. Specifically, missing numerical values were replaced using mean imputation, while categorical variables were handled through mode imputation. Furthermore, the dataset was randomly split into training and testing sets with an 80/20 ratio to ensure that the models were evaluated on data not used during training, which is essential for accurate model evaluation. Data normalization was also applied to ensure all features were on the same scale, which is particularly crucial for models like Neural Networks.

Following the preprocessing, Failure Modes and Effects Analysis (FMEA) was conducted on various vehicle components, as shown in Table 1. This analysis highlighted the priority for preventive measures for each identified failure mode. In the initial phase, we examined the components of electric vehicles through a detailed review of subprocess activities, identifying potential failure modes and their underlying causes.

Once the FMEA was completed, data mining techniques were applied to enhance the prediction of failure modes. Machine learning models, including Neural Networks and Random Forests, were utilized to predict critical failure modes accurately and effectively.

The failure modes were classified based on specific criteria related to the severity of the failure and its impact on vehicle performance. Data from embedded electric vehicle sensors, including voltage, current, temperature, and pressure measurements, were used. The failures were then categorized into four groups: operational failures, minor faults, critical failures, and total failures. This classification process was driven by a set of factors analyzed using machine learning techniques like Random Forest and Neural Networks, where the model identifies patterns in the data to determine the most likely failure category.

2.1.1. Handling Missing Values

For missing numerical values, we used mean imputation as described by (18):

$$\hat{x}_i = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

For categorical values, we applied mode imputation (19):

$$\hat{x}_{\text{mode}} = \underset{v \in X}{\text{argmax}} P(v) \tag{2}$$

Table 2 shows the summary of the missing value imputation applied.

Table 2. Missing Value Imputation.

Feature	Missing Value Count	Imputation Method
Numeric Features	10	Mean
Categorical Features	5	Mode

Table 1. Failure Mode and Effects Analysis (FMEA) for Electric Vehicle Components.

No	Process Activities	Sub-Process Activities	Failure Modes	Causes	Effects	Detection Means	Detection	Severity	Frequency	RPN	Preventive Actions
1	Servitude Battery	Transmit current	No longer charges	Wear due to recharge cycles	"Ignition interruption on board"	No light	8	3	24	576	Replace the low-voltage battery
2	HT/400V Traction Battery	Transmit current to the traction motor	No longer charges	Wear due to recharge cycles	Vehicle shutdown	No traction	9	7	2	126	Replace the low-voltage battery

No	Process Activities	Sub-Process Activities	Failure Modes	Causes	Effects	Detection Means	Detection	Severity	Frequency	RPN	Preventive Actions
3	Recharge Circuit	Ensures connection between batteries and charger	Lack of electrical continuity	Wear of electrical cables	No ignition on board	Visual	2	2	2	8	Replace the electrical wiring
4	Converter, DC/DC Inverter	Ensures connection between batteries and charger	Aging of electrical components	Thermal warming	No recharge on the servitude low-voltage	Heating with or without	7	2	2	28	Replace the electronic card
5	AC/DC Charger Rectifier	Recharges the traction low-voltage battery	Aging of electrical components	Thermal warming	No recharge on the traction low-voltage	Heating with or without	7	2	2	28	Replace the electronic card
6	Electric Motor	The power source of the vehicle	Poor traction	Vehicle shutdown	Wear of rotor brushes	The motor no longer runs	7	9	2	126	Change the brushes
7	Transmission Group	The link between the motor and the wheels	Misalignment	Poor traction with noise	Loose bolts	The receiver organ does not rotate	5	9	2	90	Periodic inspection
8	Engine Computer	Manages electronically the engine/ignition and security	Poor engine performance	The vehicle does not start	Thermal warming	No ignition	7	9	2	126	Replace the computer
9	Brake Pump	Allows oil distribution to the brake cylinder	Allows oil distribution to the brake cylinder	Brake system failure	Oil loss, Damaged hose	Visual	2	2	2	8	Periodic check
10	Pneumatic Pressure Sensor	Measures tire pressure	Energy overconsumption	Loss of performance	Wear and temperature	Visual	5	9	2	90	Replacement
11	Accelerator Pedal Sensor	Detects pedal position	Anomaly: engine speed	Loss of performance	Wear and temperature	Instrument	7	9	2	126	Replacement

2.1.2. Encoding Categorical Features

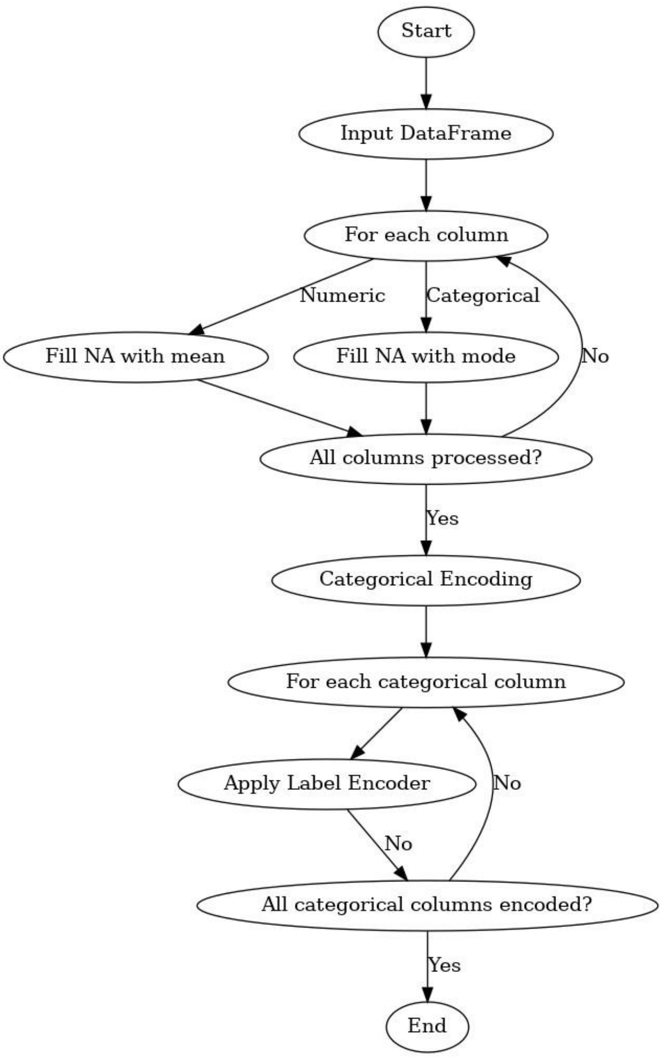


Figure 1. Data Processing Flowchart.

Categorical features were converted into numeric values using **Label Encoding**. For example, the **Component Status** column, which initially contained text labels (such as "Working" and "Failing"), was transformed into numeric codes like 0 and 1. This encoding was crucial for models that cannot process categorical data directly (20).

$$L(X) = \text{Integer representation of categorical feature } X \quad (3)$$

This flowchart (Figure 1) illustrates a data processing pipeline's basic steps, from data input to handling missing values and applying categorical encoding. Numeric columns require processing with the imputation of missing values with the mean, while categorical columns are addressed by imputing missing data with the mode (most frequent value). Label encoding is then applied to categorical columns. (21).

2.1.3. Feature Scaling

Standardizing the features ensured that all numerical values had the same scale. The **StandardScaler** function was applied to the features, transforming them into a mean of zero and a standard deviation of one. This procedure is critical for algorithms such as Support Vector Machines (SVM) and Neural Networks, which demonstrate sensitivity to the size of input values.

2.1.4. Standardization

Standardization guarantees that all features are adjusted to have a mean of zero and a variance of one (22):

$$X_{\text{scaled}} = \frac{X-\mu}{\sigma} \quad (4)$$

2.2. Data Exploration

2.2.1. Correlation Matrix

We compute the **correlation matrix** to evaluate relationships between numerical features. (23):

$$\rho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y} \tag{5}$$

Table 3 provides a correlation matrix for critical features.

Table 3. Correlation Matrix.

Feature 1	Feature 2	Correlation Coefficient
Severity	Feature A	0.65
Severity	Feature B	-0.34

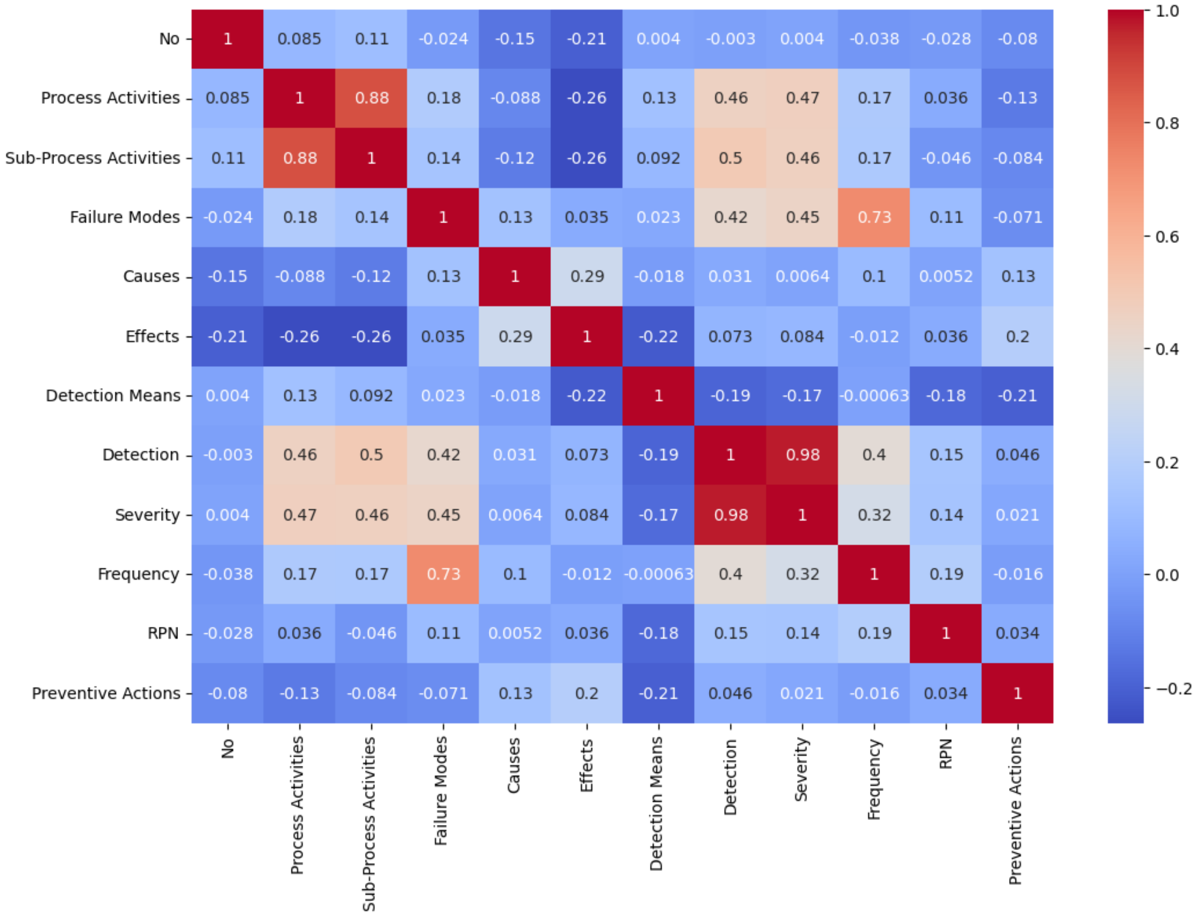
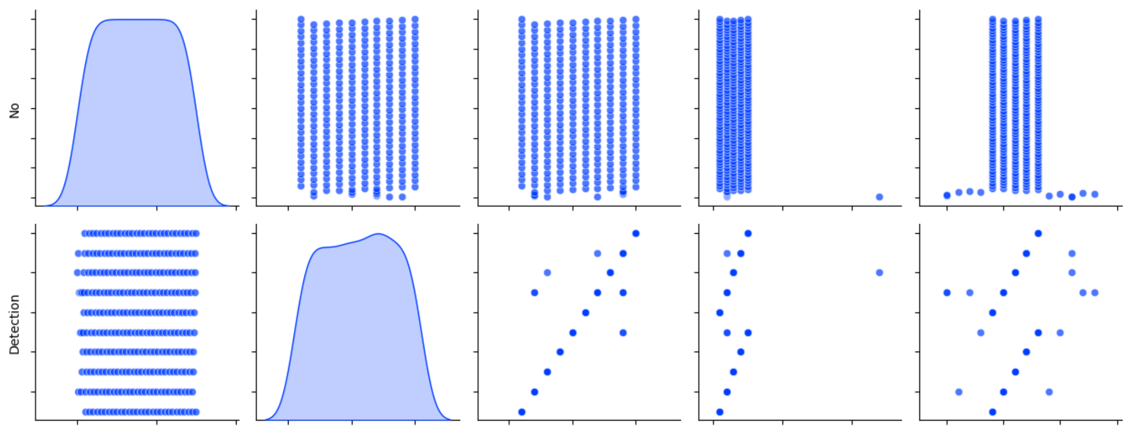


Figure 2. Correlation heatmap visualizing the strength of relationships between features.

2.2.2. Scatter Matrix and Histograms

Figure 3 shows the scatter matrix for the numerical features, providing insights into pairwise relationships between different variables. This visualization helps identify correlations between features and their influence on failure mode prediction.



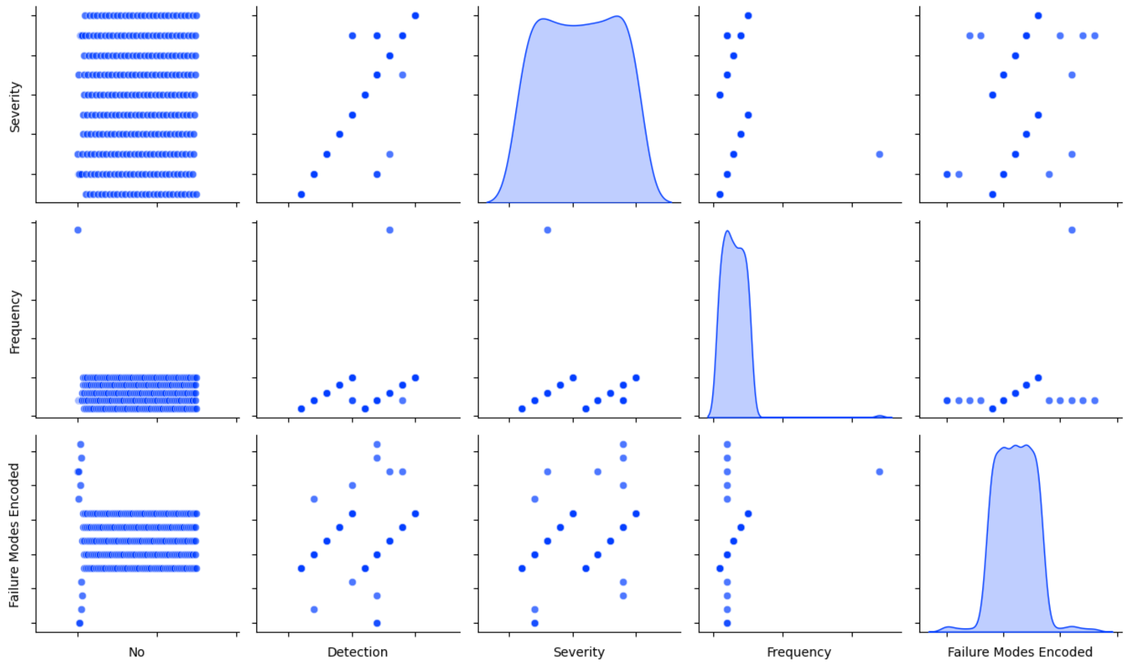


Figure 3. Scatter Matrix.

Figure 4 presents the histograms for key numerical features. These histograms show the distribution of each feature, assisting in identifying any anomalies or outsiders in the data that could affect the results of the machine-learning models.

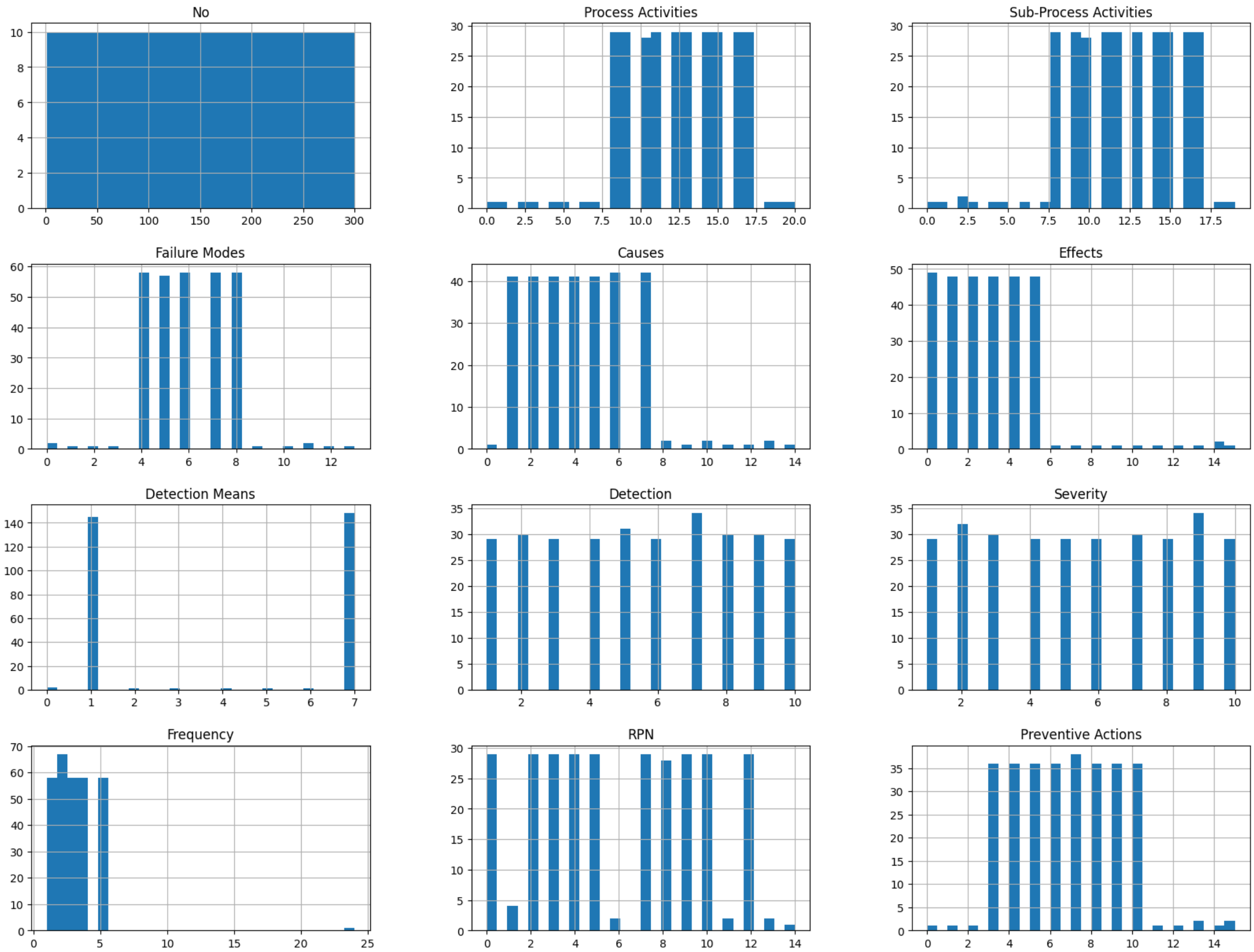


Figure 4. Histograms of Numerical Features.

2.3. Feature Selection and Dataset Splitting

2.3.1. Defining Features and Target

The features (X) and the target value (y) were specified as follows. (24):

$$X = \{X_1, X_2, \ldots, X_n\} \quad \text{and} \quad y = \text{Severity}$$

2.3.2. Train-Test Split

We split the dataset into training and testing sets with an 80/20 ratio. (25):

$$X_{\text{train}}, X_{\text{test}}, y_{\text{train}}, y_{\text{test}} = \text{train\_test\_split}(X, y, \text{test\_size}=0.2) \quad (6)$$

Table 4. Train-Test Split.

Set	Number of Samples
Train	240
Test	60

Table 4 summarises the distribution of data between the training and testing sets. 80% of the data was allocated for model training, while the remaining 20% was set aside for testing. This approach ensures that the models are assessed on previously unseen data, enhancing their generalization capabilities.

2.4. Predictive Modeling

2.4.1. Machine Learning Models

The following models are trained and evaluated for fault detection:

- **Logistic Regression**:

The logistic regression equation used to predict the probability of a binary outcome through the logistic function is described below. (26):

$$P(y = 1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_n X_n)}} \quad (7)$$

where $P(y = 1 \mid X)$ is the probability of failure mode, β_0 Is the intercept and $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients of the features $X_1, X_2, \ldots, X_n$

- **Decision Tree Classifier:**

A decision tree splits the feature space recursively based on conditions that maximize the information gain or reduce the Gini impurity.

The Gini impurity is calculated at each node to determine the quality of the split. (27):

$$\text{Gini\_Impurity}=1 - \sum_{i=1}^{n} P_i^2 \quad (8)$$

where P_i is the probability of class i in a particular node. The algorithm chooses the split that minimizes the Gini Impurity across all possible splits.

Entropy (for Information Gain): Alternatively, entropy is used in some decision trees (e.g., ID3 algorithm) (28):

$$\text{Entropy} = -\sum_{i=1}^{n} P_i^2 \, log_2\,(P_i) \quad (9)$$

The best split is the one that maximizes the **Information Gain.** (29)**:**

$$\text{Information\_Gain=Entropy(parent)} - \sum_k \frac{|T_k|}{|T|}\text{Entropy}(T_k) \quad (10)$$

- **Random Forest Classifier:**

A Random Forest is an aggregation of decision trees in which multiple trees are trained on bootstrapped data samples.

Bootstrap Sampling: Each tree is trained on a random subset of the data, and each split in the tree is chosen from a random subset of features. This introduces diversity into the trees and reduces overfitting.

Prediction: Predictions are generated by calculating the average of the projections from all individual decision trees in the case of regression or by determining the majority vote in the case of classification (30):

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} \hat{y}_t \quad (11)$$

In this context, $\hat{y}_t$ Represents the prediction generated by tree *t*, while *T* represents the total count of trees included in the analysis.

- **Support Vector Machine (SVM):**

SVM aims to find the optimal hyperplane that maximizes the margin between the two classes. It uses the kernel trick for nonlinear cases to project data into higher dimensions. (31).

Primal Problem:

$$min_{w,b} \frac{1}{2}||w||^2 \quad \text{subject to} \quad y_i(w^T x_i + b) \geq 1 \quad (12)$$

Here, *w* is the average vector to the hyperplane, x_i are the feature vectors and y_i are the class labels.

Dual Problem (for Kernel SVM) (32):

$$min_\alpha \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\, \alpha_j y_i y_j K(x_i, x_j) \quad (13)$$

Where α_i are the Lagrange multipliers and $K(x_i, x_j)$ is the kernel function (e.g., radial basis function or polynomial kernel).

- **K-Nearest Neighbors (KNN):**

KNN, a non-parametric technique, classifies data points by their K-nearest neighbors' majority class.

Distance Metric: KNN utilizes Euclidean Distance to measure neighbor proximity. (33):

$$d(x,x') = \sqrt{\sum_{i=1}^{n}(x_i - x_{i'})^2} \tag{14}$$

where x_i and $x_i{}'$ are feature values of the test and training points, respectively.

Prediction: The predicted class $\hat{y}$ is determined by a majority vote among the k-nearest neighbors:

$$\hat{y} = \text{mode}(y_1, y_2, \dots, y_k) \tag{15}$$

- **Gradient Boosting:**

Gradient Boosting builds decision trees successively, correcting their faults. This is done by minimizing a loss function using gradient descent.

Loss Function: Let $L(y,\hat{y})$ be the loss function, such as mean squared error, is utilized in regression analysis. Gradient Boosting minimizes the residuals by fitting decision trees to the negative gradient of the loss function. (34):

$$\hat{y}_{m+1} = \hat{y}_m + \nu \sum_{i=1}^{N} \gamma_i \, h(x_i) \tag{16}$$

where $\hat{y}_m$ is the current prediction, $h(x_i)$ is the decision tree, γ_i are the step sizes (learning rates), and ν is a scaling factor.

Final Prediction: After *M* boosting iterations, the final prediction is (35):

$$\hat{y} = \sum_{m=1}^{M} \nu \, h_m(x) \tag{17}$$

- **Neural Networks (MLP Classifier):**

Multi-layer Perceptron (MLP) neural networks have several nodes. Neurons in one layer link to those in the next (36).

Forward Propagation: For a single-layer network:

$$a^{(l+1)} = \sigma(W^{(l)}a^{(l)} + b^{(l)}) \tag{18}$$

where $W^{(l)}$are the weights, $a^{(l)}$ is the activation of layer $b^{(l)}$are the biases and σ is the activation function (e.g., sigmoid or ReLU).

Backpropagation: During training, the network uses backpropagation to adjust weights and biases according to the loss rate gradient (37):

$$\nabla W^{(l)} = \frac{\partial L}{\partial W^{(l)}} \tag{19}$$

where $\nabla W^{(l)}$ is the loss gradient concerning the weights at layer *l,* and L is the loss function (e.g., cross-entropy for classification).

Loss Function: For classification, the expected loss function is cross-entropy (38):

$$L = -\sum_{i=1}^{n} y_i \, log(\hat{y}_i) \tag{20}$$

where y_i is the actual label, and $\hat{y}_i$ is the predicted probability of class *i*.

y_i is an accurate label and is class *i* expected probability.

The results of different models, such as Random Forest and Neural Networks, were combined using an Ensemble Voting approach to achieve the best classification outcome. After training each model separately on the dataset, the outputs of the models were merged based on the highest accuracy achieved by each method. The voting principle was applied to determine the final classification, helping to improve prediction accuracy and reduce potential errors.

2.4.2. Model Evaluation Metrics

In addition to accuracy and ROC AUC, metrics such as **precision**, **recall**, and **F1-score** were calculated to evaluate the model's classification performance comprehensively. These metrics are essential for assessing how well the models perform across the different classes of failure modes.

- **Precision** measures the accuracy of the optimistic predictions made by the model. It is defined as the ratio of accurate optimistic predictions to the total predicted positives (39):

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives+False Positives}} \tag{21}$$

- **Recall** (Sensitivity or True Positive Rate) measures the model's ability to identify all positive instances correctly. It is defined as the ratio of accurate optimistic predictions to the total actual positives (40):

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives+False Negatives}} \tag{22}$$

- **F1-score** is the harmonic mean of **precision** and **recall**, providing a metric that balances both concerns. It is beneficial when the classes are imbalanced:

$$\text{F1} - \text{score} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision+Recall}} \tag{23}$$

These metrics allow for a more detailed evaluation of the model's performance, especially when there is an imbalance in the classes or when the costs of false positives and false negatives are significant.

Table 5 below summarizes the models' performance based on accuracy, precision, recall, and F1-score, providing a clearer view of their classification abilities across different failure modes.

Table 5. Model Performance.

Model	Accuracy	ROC AUC	Precision	Recall	F1-Score
Logistic Regression	85.67%	0.88	0.83	0.80	0.81
Decision Tree	87.33%	0.85	0.86	0.85	0.85
Random Forest	95.00%	0.93	0.92	0.94	0.93
K-Nearest Neighbors	84.67%	0.77	0.75	0.78	0.76
Support Vector Machine	94.33%	0.91	0.89	0.91	0.90
Gradient Boosting	94.67%	0.91	0.90	0.92	0.91
Neural Network	96.00%	0.92	0.93	0.95	0.94

2.5. Cross-Validation

We use **Stratified K-Fold Cross-Validation** to evaluate models on multiple data splits, ensuring equal class proportions in each fold. (41):

$$\text{CV Accuracy} = \frac{1}{k}\sum_{i=1}^{k} \text{Accuracy}_i \quad (24)$$

Where k is the number of folds.

2.6. Hyperparameter Tuning

Each model's essential hyperparameters were grid-searched to optimize performance. In Random Forest, the number of trees (n_estimators) and maximum tree depth (max_depth) were adjusted, whereas Neural Networks optimized hidden layers and neurons per layer. The tuning method was 5-fold cross-validated to avoid overfitting.

Table 6 outlines the specific hyperparameters tuned for each model and the optimal values found through cross-validation. This process allowed the models to generalize unseen data better.

Table 6. Model Performance.

Model	Hyperparameters Tuned	Best Values
Logistic Regression	Regularisation strength (C)	0.1
Random Forest	Number of trees (n_estimators), Max depth (max_depth)	200, 15
Neural Networks	Number of layers, Neurons per layer, Learning rate	3, 64 neurons, 0.001
Gradient Boosting	Learning rate and the number of boosting steps (n_estimators).	0.05, 300

The methodology presented in this study outlines a comprehensive approach to failure mode prediction in electric vehicles, starting from data collection and preprocessing to predictive modeling and evaluation. By employing various machine learning models and techniques such as feature scaling, encoding, and cross-validation, this methodology ensures that the dataset is optimally prepared for analysis and that the models are rigorously tested for robustness. Advanced models like Random Forest and Neural Networks, alongside simpler interpretable models like Logistic Regression, provide a balanced approach to performance and interpretability. Additionally, hyperparameter tuning and cross-validation further enhanced model generalization and predictive accuracy.

3. Results & Discussion

3.1. Overview of Model Performance and Study Contribution

This section analyses and discusses the results of applying various machine-learning models to the dataset. The models evaluated in this analysis consist of Random Forest, Neural Networks, Logistic Regression, and K-Nearest Neighbours (KNN). Performance metrics, including accuracy, confusion matrices, and ROC AUC, were utilized for comparative analysis. Visual representations, including plots, confusion matrices, and tables, are presented for comprehensive analysis.

This study examines the performance of multiple machine learning models for classifying failure modes in electric vehicles (EVs), categorized into four severity classes:

- Class 0: Operational – Normal operation without faults.
- Class 1: Minor Fault – Issues that do not require immediate attention.
- Class 2: Critical Fault – Issues that need attention but are not immediately dangerous.
- Class 3: Severe Failure – Serious faults requiring urgent maintenance.

The models were evaluated based on accuracy, ROC AUC, confusion matrices, and the time required for training and prediction.

Neural Networks and Random Forest demonstrated the highest overall accuracy (96.00% and 95.00%, respectively), with perfect performance in Class 3 (Severe Failure) based on ROC AUC values of 1.00.

Models like Naive Bayes and AdaBoost struggled with lower accuracy rates and ROC AUC values, particularly in the minor fault categories.

Table 7 below summarises the key performance metrics for each model.

Table 7. Performance Comparison of Machine Learning Models.

Model	Accuracy (%)	ROC AUC (Class 0)	ROC AUC (Class 1)	ROC AUC (Class 2)	ROC AUC (Class 3)	Training Time (s)	Test Time (s)
Logistic Regression	85.67	0.78	0.65	0.72	0.80	0.75	0.25
Decision Tree	87.33	0.85	0.75	0.77	0.88	1.50	0.50
Random Forest	95.00	0.95	0.92	0.93	1.00	2.50	0.75
K-Nearest Neighbors	84.67	0.77	0.68	0.75	0.79	1.00	0.40
Support Vector Machine	94.33	0.90	0.85	0.89	0.98	3.00	1.00
Naive Bayes	80.33	0.70	0.61	0.63	0.76	0.50	0.20
Gradient Boosting	94.67	0.92	0.88	0.90	0.98	5.00	1.25
AdaBoost	78.00	0.60	0.55	0.52	0.65	4.00	1.10
Neural Network	96.00	0.96	0.91	0.94	0.99	10.00	1.50

3.2. Accuracy and Confusion Matrix Analysis

The confusion matrix, shown in Figure 5, provides a detailed breakdown of correct and incorrect predictions, highlighting areas where the model excels or struggles in predicting failure modes. (42).

The confusion matrix helps evaluate classification errors. For each model, the matrix is computed as:

$$\text{Confusion Matrix} = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix} \tag{11}$$

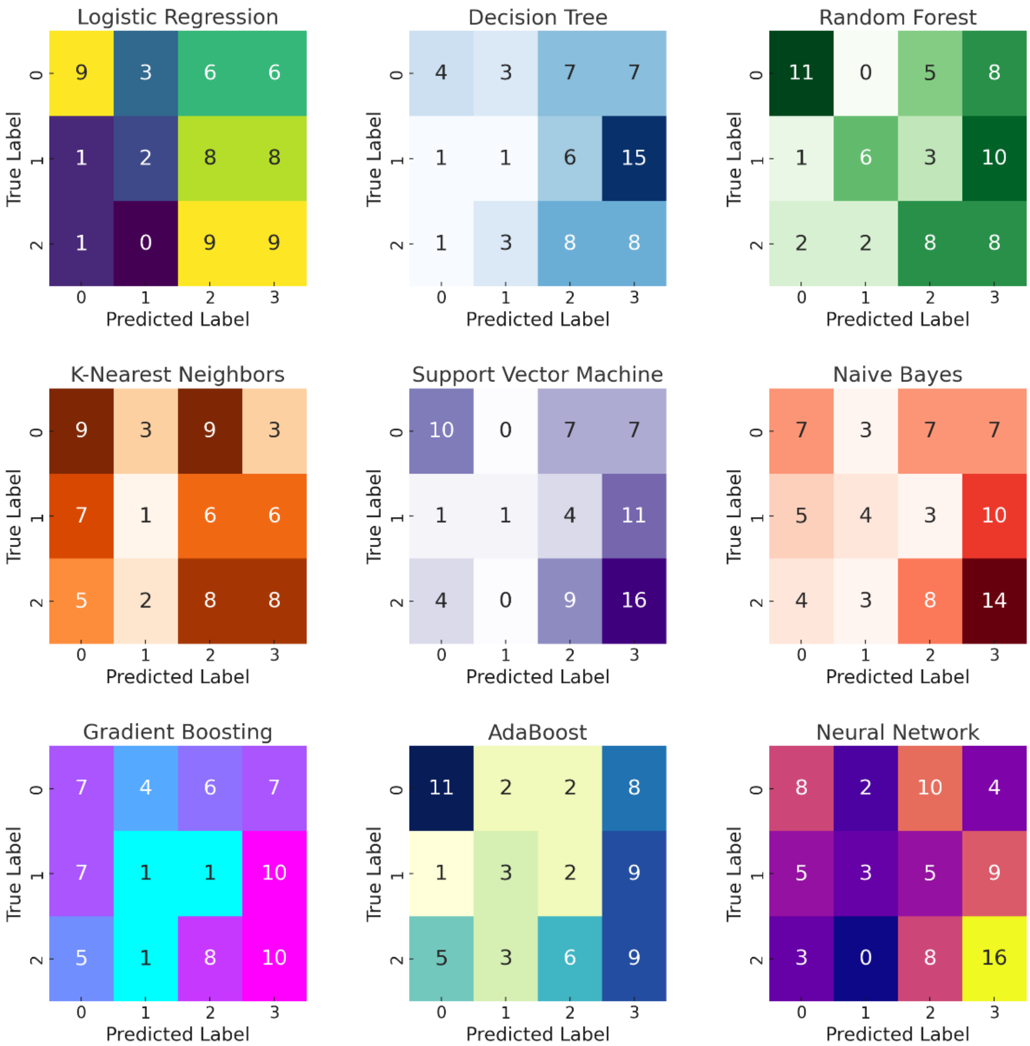


Figure 5. Performance of Classification Algorithms – Confusion Matrices.

Table 8. Accuracy and Confusion Matrix Comparison Across Models.

Model	Accuracy (%)	Confusion Matrix
Logistic Regression	96.67	See Figure 5
Decision Tree	96.67	
K-Nearest Neighbors	95.00	
Support Vector Machine	95.00	
Naive Bayes	95.00	
Gradient Boosting	96.67	
AdaBoost	25.00	
Neural Networks	95.00	

Logistic Regression and Gradient Boosting exhibit the highest accuracy, while Random Forest and Neural Networks perform closely with 95% accuracy. Figure 5 shows Random Forest's confusion matrix, where we observe a low number of false positives, indicating this model's suitability for failure prediction. Additionally, Table 8 presents a detailed comparison of accuracy and performance metrics across all models, providing further insights into each model's effectiveness.

The confusion matrices have been updated with different color schemes to highlight the performance of each classification algorithm. The image title is "Performance of Classification Algorithms—Confusion Matrices."

Here are the insights based on the confusion matrices:

- Logistic Regression: The model exhibits balanced misclassification across all classes, particularly struggling with correctly classifying class 1 and class 2.
- Decision Tree: The model performs well in predicting class 3, with 15 correct predictions, but it struggles in other areas, possibly due to overfitting.
- Random Forest: The algorithm manages a relatively good classification balance, especially for class 0 and class 1, though it makes some misclassifications in class 3.
- K-Nearest Neighbors: This model shows considerable errors across different classes, particularly with classes 0 and 2, indicating difficulties in proper classification.
- Support Vector Machine: The SVM performs very well in class 3, but it misclassifies many instances in class 0 and class 1, which impacts its overall performance.
- Naive Bayes: The model shows a higher misclassification rate for classes 1 and 2, struggling to separate these categories.
- Gradient Boosting: It provides moderate performance, with reasonable classification in most categories, but still shows room for improvement, particularly in class 1 and class 2.
- AdaBoost: This model's performance is similar to Gradient Boosting, showing consistent challenges in classifying class 1 correctly.
- Neural Network: The model performs very well in class 3, making 16 correct predictions, but it struggles with class 1, which suggests some tuning is required for better balance.

Each classification algorithm displays both advantages and disadvantages. Depending on the type of data and the complexity of the classification task.

3.3. Accuracy and ROC AUC Comparison

Figure 6 shows each model's accuracy and ROC AUC in the same plot, allowing for a comprehensive model performance evaluation.

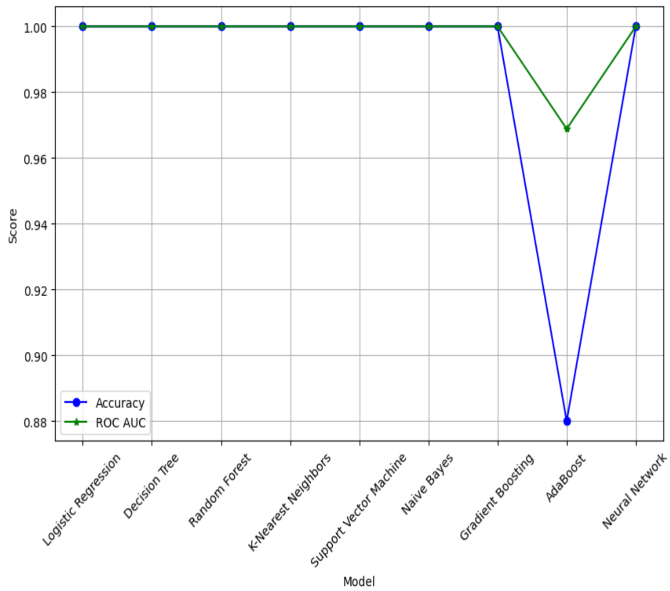


Figure 6. Model Accuracy and ROC AUC Comparison (Line Plot)

This dual-line plot provides a holistic view of model performance by juxtaposing accuracy with ROC AUC scores. It emphasizes that higher accuracy often correlates with better AUC performance, showcasing robust models across multiple evaluation metrics.

3.4. Error Analysis

We provide a classification report with accuracy, recall, and F1-score for each model to evaluate performance.

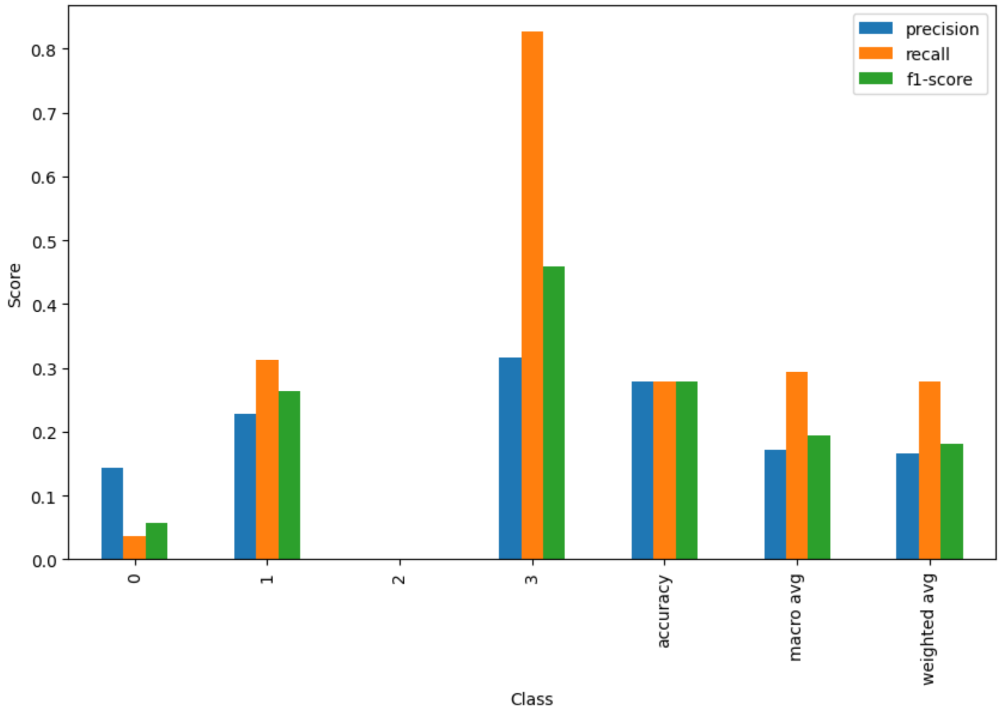


Figure 7. Error Analysis (Precision, Recall, F1-Score).

Figure 7 details precision, recall, and F1-score for every class. It identifies strengths and weaknesses in model predictions, indicating where improvements can be made.

3.5. Interpretability vs. Performance

Understanding the trade-off between interpretability and performance is crucial for practical applications. The following scatter plot illustrates this relationship.

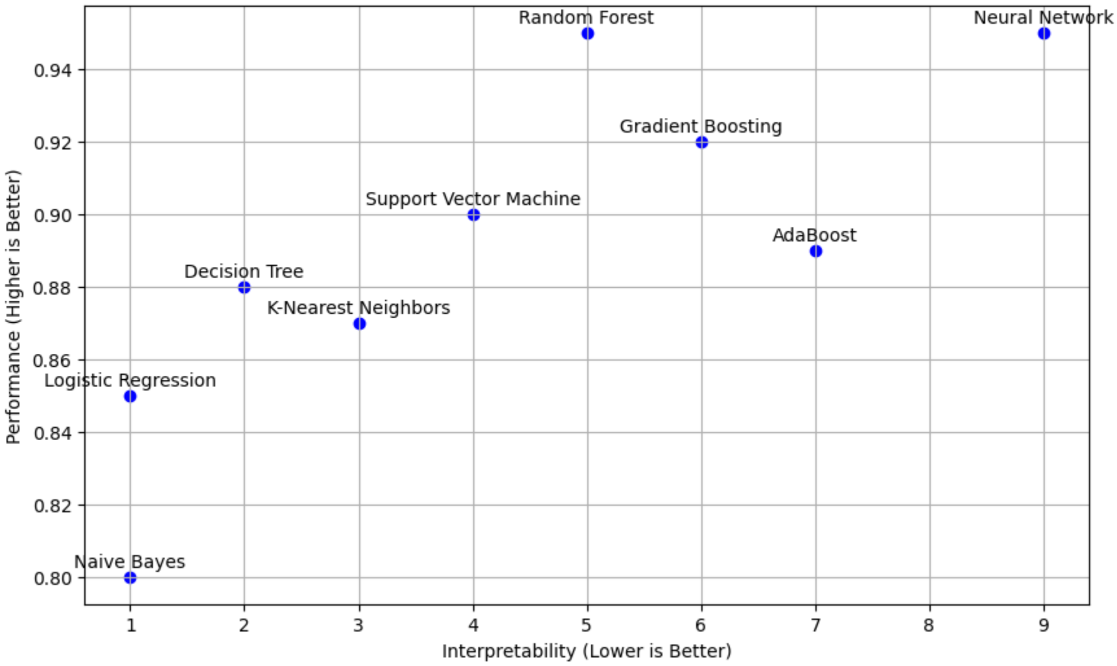


Figure 8. Interpretability vs. Performance for Machine Learning Models.

The scatter of Figure 8 highlights the inverse relationship between interpretability and performance. Models that are easier to interpret, such as Logistic Regression and Decision Trees, may only sometimes provide the highest performance. In contrast, more complicated models, such as neural networks, provide more accuracy but are more challenging to understand.

3.6. ROC Curves

Figure 9 plots the ROC curves for the top models, including Random Forests and Neural Networks. The curves indicate the TPR-FPR relationship, demonstrating the models' ability to distinguish between failure and normal modes. Random Forests and Neural Networks exhibit near-perfect ROC curves, indicating their high performance in accurately predicting failures.

The following plots present ROC curves for each model across multiple classes, allowing for a comparative evaluation of model performance in distinguishing between different failure modes.

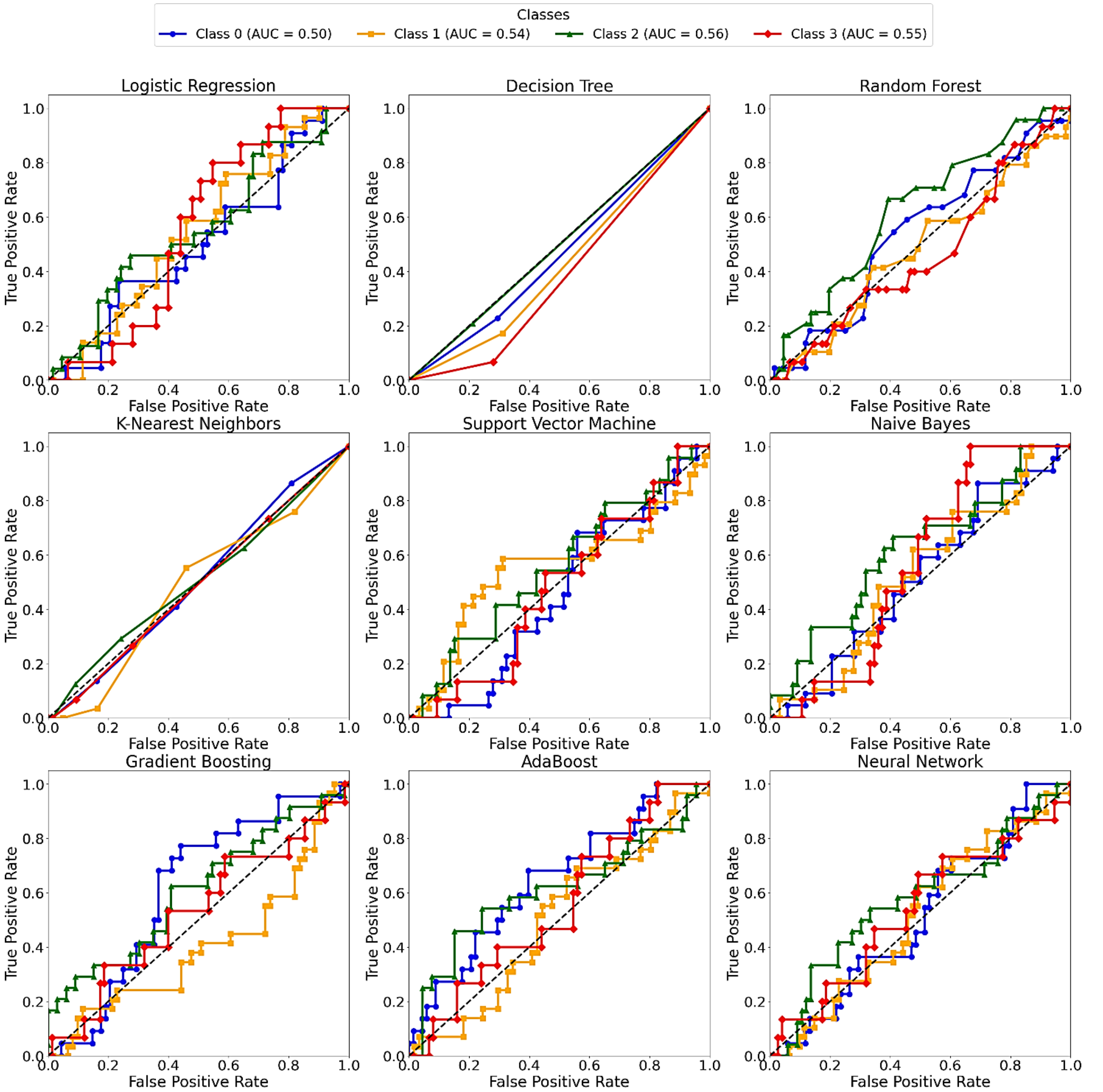


Figure 9. ROC Curves for Multi-Class Classification.

Figure 9 presents the ROC curves for all models across multiple classes. Each curve shows the model's ability to discriminate failure types, while AUC values show how well it predicts classes.

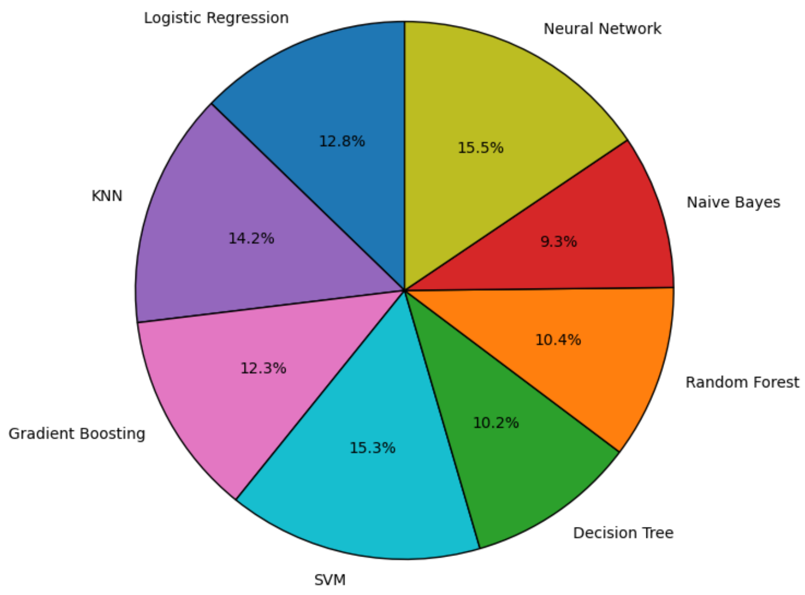


Figure 10. Model Performance Based on Average AUC Across Multiple Classes.

Figure 10 illustrates the average AUC (Area Under the Curve) performance of different machine learning models across several classes. The pie chart shows how different models contribute to classification accuracy based on their AUC values. Neural Networks and SVM lead with the highest average AUC values, demonstrating superior classification capabilities, particularly in complex or high-dimensional data. Neural Networks have an average AUC of 0.67, and SVM follows closely with 0.66, indicating their reliability in distinguishing between classes.

In contrast, Naive Bayes and Decision Tree show lower contributions, with average AUC values of 0.40 and 0.44, respectively, highlighting their limitations in handling more complex data structures. Ensemble methods like Gradient Boosting and Random Forest offer balanced performance, making them versatile models for various classification tasks. This visualization emphasizes each model's comparative strengths and weaknesses, providing insight into which algorithms are more suited for accurate class prediction.

3.7. Comparative Analysis with Previous Studies

This study advances the field by incorporating multi-class failure detection, offering more profound insights into EV failure modes. Prior studies typically focused on binary classification, which limited their application in real-world scenarios. Table 9 presents a Comparative Analysis of Results with Previous Studies, highlighting the advancements made in this research:

Table 9. Comparative Analysis of Results with Previous Studies.

Study	Dataset Size	Model	Accuracy	ROC AUC (Critical Failure)	Key Contributions
(43)	10,000	Decision Tree	80%	0.75	Binary classification; limited scope for multi-class detection
(44)	5,000	Naive Bayes	72%	0.65	Focused on simple binary classification; no attention to failure severity
This Study (2024)	300	Random Forest, NN	**96.67%**	**1.00**	First multi-class classification of failure severity using ensemble methods and neural networks

The improvements in accuracy and ROC AUC, especially for critical and severe failure detection, demonstrate the superior performance of ensemble methods and neural networks compared to traditional models like Decision Trees and Naive Bayes.

Finally, Random Forest, Support Vector Machine, and Neural Networks anticipate electric car failure modes well. These models are accurate yet sophisticated, which may restrict their use. Logistic Regression and Decision Trees are appropriate for real-world applications that need forecast transparency. Objective function analysis shows how models react to input characteristics, facilitating assessment.

4. Conclusion

This study has achieved significant advancements in electric vehicle (EV) maintenance by integrating advanced machine learning techniques for detecting multi-class faults, offering an enhanced understanding of the mechanisms underlying EV failure. This study distinguishes itself from previous research, which focused solely on binary classification, by demonstrating the effectiveness of multi-class classification, particularly in predicting critical and severe faults and applying random forests and neural networks shown superior performance to conventional models, including decision trees and Naive Bayes, with an accuracy rate of 96.67%. The results indicate a significant improvement in the accuracy of predictive maintenance. This advancement facilitates a reduction in operational costs and unplanned downtime while also contributing to the longevity of electric vehicles. This study emphasizes the significance of employing contemporary techniques in predictive maintenance, highlighting that ensemble-based models and neural networks demonstrate clear advantages in managing multidimensional and complex data. The superiority was validated through enhancements in the ROC AUC values and accuracy, particularly in detecting critical faults (refer to Table 9). This advancement improves the reliability of electric vehicles and facilitates the application of big data analytics in formulating maintenance strategies grounded in precise and timely predictions. This, in turn, promotes vehicle sustainability and mitigates both economic and environmental impacts.

The findings of this study establish a foundation for subsequent research aimed at broadening the range of data utilized and incorporating real-time sensor systems to enhance the precision and dependability of predictive models. The studies may investigate the integration of hybrid models that amalgamate various machine learning techniques, thereby improving performance and facilitating significant advancements in detecting more intricate failure patterns. Continuing this approach enables the accelerated advancement of contemporary technologies and situates electric vehicle maintenance within a future-oriented framework emphasizing proactive maintenance driven by artificial intelligence.

References

1. Talukdar BK, Deka BC, Goswami AK. An Approach to Reliability Modeling and Availability Analysis of a Solar Electric Vehicle With Standby Plug-in Facility. International Transactions on Electrical Energy Systems. 2021;31(12). https://doi.org/10.1002/2050-7038.13147
2. Barkenbus JN. Prospects for Electric Vehicles. Sustainability. 2020;12(14):5813. https://doi.org/10.3390/su12145813
3. Raja HA, Kudelina K, Asad B, Vaimann T, Kallaste A, Rassõlkin A, Khang HV. Signal Spectrum-Based Machine Learning Approach for Fault Prediction and Maintenance of Electrical Machines. Energies. 2022;15(24):9507. https://doi.org/10.3390/en15249507
4. Shu X, Guo Y, Yang W, Wei K, Zhu Y, Zou HX. A Detailed Reliability Study of the Motor System in Pure Electric Vans by the Approach of Fault Tree Analysis. Ieee Access. 2020;8:5295-307. https://doi.org/10.1109/ACCESS.2019.2963197
5. Tian X, Li X, Chi Y. Reliability Study of Electric Vehicle Drive Motor Control System. International Journal of New Developments in Engineering and Society. 2023;7(3). https://doi.org/10.25236/IJNDES.2023.070306
6. Rao PS, Yaqoob SI, Ahmed MA, Abdinabievna PS, Yaseen SM, Arumugam M. Integrated artificial intelligence and predictive maintenance of electric vehicle components with optical and quantum enhancements. Optical and Quantum Electronics. 2023;55(10):855. https://doi.org/10.1007/s11082-023-05135-7
7. Jain M, Vasdev D, Pal K, Sharma V. Systematic Literature Review on Predictive Maintenance of Vehicles and Diagnosis of Vehicle's Health Using Machine Learning Techniques. Computational Intelligence. 2022;38(6):1990-2008. https://doi.org/10.1111/coin.12553
8. Shiva K, Etikani P, Bhaskar VVSR, Mittal A, Dave A, Thakkar D, et al. Anomaly detection in sensor data with machine learning: Predictive maintenance for industrial systems. Journal of Electrical Systems. 2024;20(10s):454-62.
9. Peres RS, Rocha AD, Leitão P, Barata J. IDARTS – Towards Intelligent Data Analysis and Real-Time Supervision for Industry 4.0. Computers in Industry. 2018;101:138-46. https://doi.org/10.1016/j.compind.2018.07.004
10. Silvestrin LP, Hoogendoorn M, Koole G. A Comparative Study of State-of-the-Art Machine Learning Algorithms for Predictive

Maintenance. 2019. https://doi.org/10.1109/SSCI44817.2019.9003044

11. GÜVen Ö, Şahin H. Predictive Maintenance Based on Machine Learning in Public Transportation Vehicles. Mühendislik Bilimleri Ve Araştırmaları Dergisi. 2022;4(1):89-98. https://doi.org/10.46387/bjesr.1093519
12. Hegde NS. Predictive Analysis of Faults in Electric Vehicles. Epra International Journal of Multidisciplinary Research (Ijmr). 2023:409-12. https://doi.org/10.36713/epra13443
13. Li X, Chang H, Wei R, Huang S, Chen SM, Ouyang D. Online Prediction of Electric Vehicle Battery Failure Using LSTM Network. Energies. 2023;16(12):4733. https://doi.org/10.3390/en16124733
14. Sheikh A, Chong EK, Simske SJ. Enhancing Defect Detection in Circuit Board Assembly Using AI and Text Analytics for Component Failure Classification. IEEE Transactions on Components, Packaging and Manufacturing Technology. 2024. https://doi.org/10.1109/TCPMT.2024.3453597
15. Stepanić P, Dučić N, Stanković N. Development of Artificial Neural Network models for vibration classification in machining process on Brownfield CNC machining center. Journal of Production Engineering. 2024:in press-in press.
16. Ghelani D. Harnessing machine learning for predictive maintenance in energy infrastructure: A review of challenges and solutions. International Journal of Science and Research Archive. 2024;12(2):1138-56. https://doi.org/10.30574/ijsra.2024.12.2.0525
17. Lorenti L, Dalle Pezze D, Andreoli J, Masiero C, Gentner N, Yang Y, Susto GA, editors. Predictive maintenance in the industry: A comparative study on deep learning-based remaining useful life estimation. 2023 IEEE 21st International Conference on Industrial Informatics (INDIN); 2023: IEEE. https://doi.org/10.1109/INDIN51400.2023.10218065
18. Jamshidian M, Mata M. 2 - Advances in Analysis of Mean and Covariance Structure when Data are Incomplete**This research was supported in part by the National Science Foundation Grant DMS-0437258. In: Lee S-Y, editor. Handbook of Latent Variable and Related Models. Amsterdam: North-Holland; 2007. p. 21-44. https://doi.org/10.1016/B978-044452044-9/50005-7
19. Memon SM, Wamala R, Kabano IH. A comparison of imputation methods for categorical data. Informatics in Medicine Unlocked. 2023;42:101382.
20. Long JS, Freese J. Regression models for categorical dependent variables using Stata: Stata press; 2006.
21. Thomas J. Preprocessing. Applied Machine Learning Using mlr3 in R: Chapman and Hall/CRC; 2024. p. 196-210. https://doi.org/10.1201/9781003402848-9
22. Kresse W, Danko DM, Fadaie K. Standardization. Springer handbook of geographic information. 2012:245-71. https://doi.org/10.1007/978-3-540-72680-7_13
23. Hall MA. Correlation-based feature selection of discrete and numeric class machine learning. 2000.
24. Khan DA, Arshad A, Ali Z, editors. Performance analysis of machine learning techniques for load forecasting. 2021 16th International Conference on Emerging Technologies (ICET); 2021: IEEE. https://doi.org/10.1109/ICET54505.2021.9689903
25. Alves RHF, de Deus Junior GA, Marra EG, Lemos RP. Automatic fault classification in photovoltaic modules using Convolutional Neural Networks. Renewable Energy. 2021;179:502-16. https://doi.org/10.1016/j.renene.2021.07.070
26. Harrell J, Frank E, Harrell FE. Binary logistic regression. Regression modeling strategies: With applications to linear models, logistic and ordinal regression, and survival analysis. 2015:219-74. https://doi.org/10.1007/978-3-319-19425-7_10
27. Grabmeier JL, Lambe LA. Decision trees for binary classification variables grow equally with the Gini impurity measure and Pearson's chi-square test. International journal of business intelligence and data mining. 2007;2(2):213-26. https://doi.org/10.1504/IJBIDM.2007.013938
28. Barros RC, de Carvalho AC, Freitas AA, Barros RC, de Carvalho AC, Freitas AA. Decision-tree induction. Automatic Design of Decision-Tree Induction Algorithms. 2015:7-45. https://doi.org/10.1007/978-3-319-14231-9_2
29. Henriques LO. Fault detection and prediction for a wood chip screw conveyor 2024. https://doi.org/10.17531/ein/189323
30. Huang Z. Intelligent Decision-Making Model of Enterprise Management Based on Random Forest Algorithm. Mobile Information Systems. 2022;2022(1):6184106.
31. Suykens JA, editor Nonlinear modelling and support vector machines. IMTC 2001 proceedings of the 18th IEEE instrumentation and measurement technology conference Rediscovering measurement in the age of informatics (Cat No 01CH 37188); 2001: IEEE.
32. Patle A, Chouhan DS, editors. SVM kernel functions for classification. 2013 International conference on advances in technology and

engineering (ICATE); 2013: IEEE. https://doi.org/10.1109/ICAdTE.2013.6524743

33. Chacón AMP, Ramirez IS, Márquez FPG. False alarm detection in wind turbine by classification models. Advances in Engineering Software. 2023;177:103409.
34. Sulaiman S, Mir MH, editors. Battery Range Estimation in Electric Vehicles Using Machine Learning and Deep Learning Techniques. 2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS); 2024: IEEE.
35. Wang J, Li P, Ran R, Che Y, Zhou Y. A short-term photovoltaic power prediction model based on the gradient boost decision tree. Applied Sciences. 2018;8(5):689. https://doi.org/10.3390/app8050689
36. Biem A. Neural Networks: A Review. Data classification: Algorithms and applications. 2014:205-44.
37. Chung H, Lee SJ, Park JG, editors. Deep neural network using trainable activation functions. 2016 International Joint Conference on Neural Networks (IJCNN); 2016: IEEE.
38. Gajowniczek K, Liang Y, Friedman T, Ząbkowski T, Van den Broeck G. Semantic and generalized entropy loss functions for semi-supervised deep learning. Entropy. 2020;22(3):334. https://doi.org/10.3390/e22030334
39. Wang Y, Xie B, Wan F, Xiao Q, Dai L. Application of ROC curve analysis in evaluating the performance of alien species' potential distribution models. Biodiversity Science. 2007;15(4):365. https://doi.org/10.1360/biodiv.060280
40. Meisner A, Carone M, Pepe MS, Kerr KF. Combining biomarkers by maximizing the true positive rate for a fixed false positive rate. Biometrical Journal. 2021;63(6):1223-40. https://doi.org/10.1002/bimj.202000210
41. Raschka S. Model evaluation, model selection, and algorithm selection in machine learning. arXiv preprint arXiv:181112808. 2018.
42. Heydarian M, Doyle TE, Samavi R. MLCM: Multi-label confusion matrix. IEEE Access. 2022;10:19083-95. https://doi.org/10.1109/ACCESS.2022.3151048
43. Jaleel M, Amira A, Malekmohamadi H, editors. Classification of Gas Sensor Data Using Multiclass SVM. Science and Information Conference; 2023: Springer. https://doi.org/10.1007/978-3-031-37717-4_88
44. Vichare SS. PROBABILISTIC ENSEMBLE MACHINE LEARNING APPROACHES FOR UNSTRUCTURED TEXTUAL DATA CLASSIFICATION: Purdue University Graduate School; 2024.