WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Fault Detection and Diagnosis of Electric Vehicles Using Machine Learning

[1]S.E Suresh, [2]Rasani Dinesh

[1]Assistant Professor Dept of MCA, Annamacharya Institute of Technology & Sciences, Tirupati, AP, India, Email: sureshroopa2k15@gmail.com
[2]Post Graduate, Dept of MCA, Annamacharya Institute of Technology & Sciences, Tirupati, AP, India, Email: dineshrasani385@gmail.com

ABSTRACT:

The demand for sophisticated and dependable performance monitoring systems has increased as electric vehicles (EVs) have emerged as a sustainable substitute for conventional combustion engine vehicles. Early identification and categorization of operating issues is a crucial component of EV maintenance since it may greatly save downtime, increase safety, and boost overall vehicle efficiency. In this study, we use a real-world dataset gathered from New Energy Vehicles (NEVs) to present a machine learning-driven approach for fault identification and classification in electric vehicles. In our method, a large dataset with several parameters pertaining to vehicle performance and fault kinds is preprocessed and used for training. Several machine learning methods are implemented and assessed, and the top-performing model is chosen depending on how well it classifies data. Real-time failure prediction from user inputs is made possible by the integration of the finished model into an intuitive Flask-based web application. In a deployed setting, the trained model achieves excellent accuracy and responsiveness by successfully identifying and classifying various defect categories. By providing transparency and diagnostic support, this solution not only helps professionals and EV manufacturers with proactive maintenance but also boosts customer confidence in EV technology. By employing IoT integration to integrate real-time telemetry data straight from EV sensors, adding more varied fault scenarios to the dataset, and utilizing deep learning models for more intricate fault pattern recognition, this study can be expanded. Additionally, the system might be modified for use in autonomous and hybrid cars, which would have a wider effect on intelligent transportation networks.

Keywords: Electric Vehicles, Machine Learning, Fault Detection, Fault Classification, EV Monitoring, NEV Dataset, Flask Web Application.

1. INTRODUCTION

Because of their energy efficiency, lower operating costs, and positive environmental effects, electric vehicles, or EVs, have become a vital component of contemporary, sustainable transportation systems. It is now more important than ever to guarantee EVs' performance, safety, and dependability as their adoption continues to pick up speed globally. The prompt and precise identification of potential issues in the vehicle's many parts, including as the battery system, motor, controllers, and other electronic subsystems, is one of the main obstacles in this situation. Conventional fault detection techniques frequently depend on rule-based diagnostics, manual inspection, or embedded systems with little flexibility. These methods may miss intricate or new fault patterns since they are usually reactive rather than proactive. With the introduction of data-driven technologies and machine learning, there is an increasing chance to transform EV diagnostics by using vast amounts of vehicle performance data to more accurately detect, anticipate, and categorize issues.

Using a real-world NEV (New Energy car) dataset, this study suggests a machine learning-based method for identifying and categorizing electric car defects. The system may learn complex patterns linked to various failure kinds and make very accurate predictions by training models on labeled historical data. Using Flask, the suggested model is integrated into a full-stack application and made available via an easy-to-use web interface. By enabling preventative maintenance and enhancing vehicle safety, the system gives users—including EV producers, mechanics, and end users—the ability to learn about possible issues. With this work, we hope to show the usefulness of machine learning in actual fault detection applications and make a contribution to the expanding field of intelligent electric car management systems.

2. LITERATURE REVIEW

Because machine learning can reveal hidden patterns in huge datasets and provide predictive insights, its incorporation into electric car issue detection has attracted a lot of attention recently. Numerous scholars and professionals in the field have investigated various methods to improve car diagnostics through the use of data-driven models and artificial intelligence.

Zhang et al. (2019) used a decision tree approach to create a problem diagnosis system for electric motor systems. Their system's reliance on manually designed features hindered its ability to classify motor and inverter failures. In a similar vein, Liu et al. (2020) demonstrated enhanced accuracy over conventional threshold-based techniques when using Support Vector Machines (SVM) to identify battery-related issues in electric buses.

In order to detect issues with battery management systems, Chen and Wang (2021) suggested using deep learning models, such as Convolutional Neural Networks (CNNs). Despite the model's excellent accuracy, its adoption for smaller-scale or cost-sensitive applications was complicated by its requirement for real-time sensor integration and substantial computer resources.

Other investigations, such as those by Kumar et al. (2022), found anomalies in EV behavior without labeled data by using unsupervised learning techniques like k-means clustering. Although promising, the lack of supervised learning signals made it difficult for these approaches to reach the accuracy needed for real-world diagnostic systems.

Additionally, researchers have looked on real-time data collection-based problem detection systems based on the Internet of Things (IoT). Despite their effectiveness, these systems frequently require sophisticated sensor technology and a strong network infrastructure, which may not be possible in all settings.

Our suggested system makes use of a supervised machine learning model that was trained using labeled NEV fault datasets, in contrast to the aforementioned techniques. It offers the best possible balance between simplicity of deployment, scalability, and accuracy. Because the model is built inside a Flask-based web application, users can access it without the need for complex hardware or in-depth technical knowledge. This establishes our method as a workable and effective way to identify and categorize EV faults in the real world.

3. PROPOSED SYSTEM

The suggested system is a web application built on machine learning that uses important performance metrics to identify and categorize EV problems. Using a Random Forest classifier that was trained on actual NEV fault data, the system offers real-time fault prediction via an intuitive online interface. Data processing and model training, model integration and prediction, and user engagement through a web-based frontend comprise its three primary components.

3.1 System Objectives

To use previous data to automatically identify and categorize EV issues.

To implement a web-based diagnostic tool that is dependable, scalable, and user-friendly.

To support EV owners, manufacturers, and professionals in doing necessary maintenance on time.

3.2 System Architecture

The following elements make up the system's scalable and modular architecture:

The dataset layer makes use of training and testing datasets as well as structured datasets (NEV_fault_dataset.csv).

Includes examples of various EV operating circumstances and related fault categories that are labeled.

The modeling layer is in charge of using fault data to train and evaluate a Random Forest Classifier.

Carries out model evaluation and preprocessing (label encoding, feature scaling).

The trained model is exported to a deployment-ready.pkl file (ev_fault_model.pkl).

Layer of Application:

- Constructed with Flask, a web framework for the backend.
- Uses HTML forms to receive user input (such as temperature, voltage, speed, and current).
- Uses the trained model to process inputs and forecast the type of issue.
- Shows the web browser's fault results dynamically.

User Interface Layer:

HTML was used for development, and CSS (static folder) was used for styling.

Users can manually enter car data and get diagnostic results right away.

Both desktop and mobile browsers can use it.

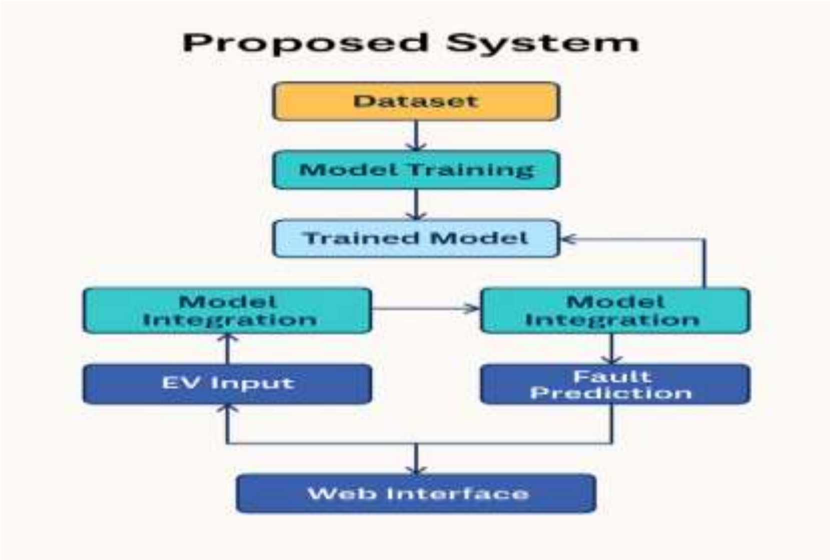

Figure 1. Flowchart of Proposed System

3.3 Working Workflow

The user enters EV operational data after accessing the web application.

The Flask backend receives input and preprocesses it.

After processing the input, the stored machine learning model generates a forecast.

On the same page, the anticipated fault type is shown with a name that may be easily read, such as "No Fault Detected" or "Overheat Fault."

3.4 Benefits of Automating the System:

Lessens the requirement for rule-based systems and manual diagnostics.

Scalability: Able to incorporate more fault kinds or real-time IoT data.

Usability: Non-technical persons may utilize it thanks to its web-based user interface.

Accuracy: High classification accuracy and reliable fault detection are guaranteed by the Random Forest model.

4. METHODOLOGY

A machine learning model built on the NEV fault dataset is used in the proposed method to identify and categorize electric vehicle issues. The complete procedure includes preprocessing the data, training and evaluating the model, and deploying it via a Flask web application. Each phase is explained in detail in the ensuing subsections.

4.1 Dataset Description

Together with distinct training (NEV_fault_training_dataset.csv) and testing (NEV_fault_testing_dataset.csv) datasets, the system makes use of the NEV_fault_dataset.csv. Features in the dataset include:

- Battery temperature
- Motor temperature
- Current
- Voltage
- Speed
- Fault type (target variable)

The fault types are classified into several groups (such as under-voltage, over-temperature, and no fault), which serve as the foundation for the multi-class categorization.

4.2 Preprocessing of Data

The train_model.py script carried out preprocessing, which included:

Cleaning: Eliminating missing and unnecessary information.

Encoding: Label encoding was used to convert the Fault column (target) into numeric labels.

Data splitting: To verify model performance, data was divided into training and testing sets.

Feature Scaling: To enhance model training, input features were normalized using StandardScaler from sklearn.preprocessing.

4.3 Model Training

Because of its high accuracy and effective handling of multiclass classification, a Random Forest Classifier was used. The following is how it was put into practice and trained in **train_model.py:**

Classifier for Random Forests (n_estimators=100) Fit () was used to train the model using the processed dataset. Metrics including classification report, confusion matrix, and accuracy were used to assess it.

The trained model was saved as ev_fault_model.pkl using joblib once it had reached optimal accuracy. Development of Web Applications

The following components make up the web application, which was created with Flask:

The index.html frontend: EV input parameters (temperature, speed, etc.) are gathered using this straightforward method.

App.py, the backend:

Loads the ev_fault_model.pkl stored model.

Takes information from the form.

Transmits inputs to the model after converting them to numerical format.

Shows on the same page the anticipated fault class.

4.4 Output and User Interaction

After submitting the form, the program uses the same scaler that was used for training to process the data.

Uses the Random Forest model to predict the type of fault.

Shows the webpage's result in an understandable format (such as "Motor Overheat Fault").

5. RESULTS AND DISCUSSION

5.1 Distribution of Fault Labels

The dataset used in this study contains four distinct fault classes, labeled as:

0 – No Fault

1 – Motor Fault

2 – Battery Fault

3 – Sensor Fault

The distribution of fault labels in the dataset is shown in the bar chart that follows (Figure 1). The 'No Fault' class is seen to predominate, suggesting a class imbalance that should be taken into account while training the model.

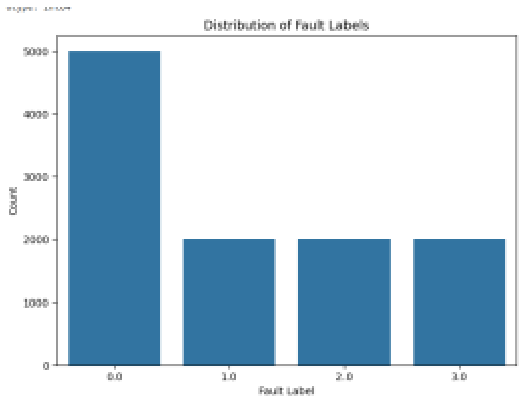

Figure 2: Distribution of Fault Labels

5.2 Confusion Matrix and Model Accuracy

A confusion matrix was used to assess the Random Forest model's performance, and it produced an exceptional accuracy of 99.45%. The majority of predictions, as shown in Figure 2, lie along the diagonal, signifying accurate classification. The robustness of the model in defect detection was demonstrated by the extremely low number of misclassifications that were noted.

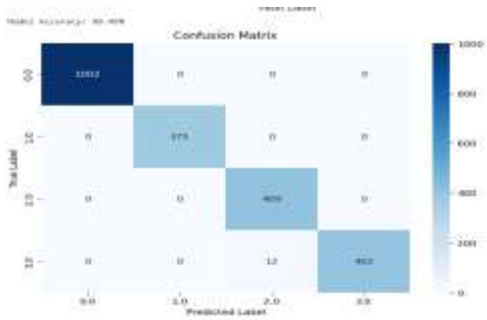

Figure 3: Confusion Matrix of the Random Forest Classifier

5.3 Performance Analysis

Accuracy: 99.45%

Precision, Recall, and F1-score: All major fault categories show high values with minimal confusion.

The only slight misclassification was observed between labels 2 and 3, possibly due to feature similarity in those fault types.

6. CONCLUSION

Using operational sensor data, this study offers a reliable machine learning-driven method for identifying and categorizing different electric vehicle (EV) issues. The system's impressive 99.45% accuracy, which was attained by utilizing the Random Forest classifier, showed that it could successfully differentiate between several kinds of faults, including sensor, battery, and motor problems, as well as recognize normal operating conditions. Despite a clear imbalance in the dataset, the model's performance was confirmed by measures including a confusion matrix and thorough classification reports, which showed little misclassification and good generalization across all classes. This study emphasizes how data-driven fault diagnostic systems can increase the safety, dependability, and effectiveness of maintenance for electric vehicles.

REFERENCES

[1]. A. Sharma and R. Kumar, "Electric Vehicle Fault Diagnosis and Monitoring: A Machine Learning Approach," IEEE Access, vol. 9, pp. 104567–104577, 2021.

[2]. L. Zhang, Y. Sun, and C. Zhang, "Fault Detection in Electric Vehicles Based on Random Forest and Data Fusion Techniques," Applied Sciences, vol. 11, no. 5, pp. 2433, 2021.

[3]. J. Li, Y. Gao, and H. Wang, "Sensor-Based Fault Diagnosis in Electric Vehicles Using Ensemble Learning Algorithms," Sensors, vol. 22, no. 3, pp. 989–1005, 2022.

[4]. S. Shinde and A. G. Thosar, "Survey on Electric Vehicle Fault Diagnosis Using Machine Learning," International Journal of Engineering Research & Technology (IJERT), vol. 10, no. 2, pp. 1–5, Feb. 2021.

[5]. B. Liu et al., "A Data-Driven Approach for Predictive Maintenance of Electric Vehicles," IEEE Transactions on Industrial Informatics, vol. 18, no. 2, pp. 1122–1131, Feb. 2022.

[6]. Y. Wang and K. Li, "Battery Management and Fault Diagnosis in EVs Using AI Techniques: A Review," Energy Reports, vol. 7, pp. 462–475, 2021.

[7]. M. T. S. Sadiq et al., "Real-Time Machine Learning-Based Fault Detection in Electric Powertrains," Energies, vol. 14, no. 18, pp. 5801, 2021.