Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

AI-Powered Fault Prediction and Optimization in New Energy Vehicles (NEVs) for the US Market

Md Sazzad Hossain[1], MD Rashed Mohaimin[2], Shah Alam[3], Md Anisur Rahman [4], Md Raisul Islam[5], Farhana Rahman Anonna[6] and Rabeya Akter[7]

[12]MBA in Business Analytics, Gannon University, Erie, PA, USA
[3]Master of Science in Information Technology, Washington University of Science and Technology, Alexandria, VA, USA
[45]Engineering Technology, Western Illinois University, Macomb, IL-61455
[67]Master of science in information technology. Washington University of Science and Technology, USA
Corresponding Author: Md Sazzad Hossain, **E-mail**: hossain005@gannon.edu

| ABSTRACT

The automotive industry in the USA is going through some significant transformation as global efforts to mitigate climate change and diminish greenhouse gas emissions intensify. Focal to this Paradigm shift is the advancement of New Energy Vehicles (NEVs), which comprise electric vehicles (EVs), plug-in hybrid electric vehicles (PHEVs), and hydrogen fuel cell vehicles (FCEVs). This research project aimed to examine the deployment of AI in forecasting and optimizing fault management in NEVs. This study intended to leverage machine learning algorithms with data analytics to provide high reliability and operational efficiency within the US automotive industry with NEVs. The dataset for the present study was accessed from accredited automotive manufacturing companies. The dataset was designed to predict the faults and optimize maintenance at NEVs. It covered simulated real vehicle data, such as sensor readings, environmental factors, driving patterns, and maintenance logs needed to understand performance, diagnose faults, and optimize a vehicle's maintenance schedule. Different algorithms were selected, such as Random Forest Classifier, Gradient Boosting Classifier, and Logistic Regression with other advantages, depending on the dataset's characteristics and the problem's complexity. Performance evaluation of the model was done with several metrics, most notably precision, recall, and F1-score. The results demonstrated that the Random Forest model attained the highest accuracy, followed closely by Gradient Boosting. AI-driven fault prediction models brought into play would greatly raise the level of impact that can be caused to the automotive industry in the US concerning the enhancement of NEV reliability and efficiency. Interpretation of the model's predictions is important in fault management strategies because it converts raw predictive outputs to actionable insights.

| KEYWORDS

Fault Prediction; New Energy Vehicles (NEV); Fault Optimization; Artificial Intelligence; US Automotive Market; Machine Learning

| ARTICLE INFORMATION

ACCEPTED: 12 December 2024 **PUBLISHED:** 06 January 2025 **DOI:** 10.32996/jcsts.2025.7.1.1

Introduction

Background

According to Shil et al. (2024), the automotive sector is going through some substantial transformation as global efforts to combat climate change and reduce greenhouse gas emissions intensify. Prime to this shift is the advancement of New Energy Vehicles (NEVs), which comprise electric vehicles (EVs), plug-in hybrid electric vehicles (PHEVs), and hydrogen fuel cell vehicles (FCEVs). NEVs are a cornerstone in the movement towards sustainable transportation: they reduce the dependence on fossil fuels and lower carbon emissions. Governments worldwide have implemented policies and incentives to accelerate the adoption of NEVs, such as tax credits, subsidies, and stricter emissions standards.

Dubois et al. (2022), reported that exponential growth in the US market for NEVs is induced by continuous developments related to battery technology, growing consumer awareness, and a rising charging infrastructure. Yet, with more NEVs on the road, expectations are on the rise: NEVs have to be much more reliable and powerful despite all the technological challenges arising from their concept. Thus, addressing the issues related to faults in NEVs has become an urgent necessity for car manufacturers and other stakeholders involved in the US automotive sector. Today's consumers are becoming very selective, looking for a vehicle that is not only "green" but also one that delivers reliable performance. In NEVs, reliability becomes an especially critical issue because all defects and problems affecting customer satisfaction tend to strongly impact brand loyalty. Therefore, car manufacturers have to develop strong automobiles that can withstand daily wear and tear with minimal maintenance costs and loss of time (Cheliah et al., 2023).

Challenges

Mamatha et al. (2024), argued that despite all the relative advantages, NEVs develop many faults that could render the performance of the motor vehicle unsatisfactory; likewise, customer satisfaction in its use is reduced to a minimum. Common failures that occur in NEVs include degradation of the traction batteries, failure of the charging system, and software problems; these may affect the functions of the vehicle negatively. Besides reduced performance, a rise in maintenance costs as well as reduced consumer confidence in the NEV technology are common repercussions from these faults. The challenge in fault detection and resolution within NEV systems is that these usually consist of complex integrations involving hardware and software.

Conventional fault detection and regulation techniques, which frequently depend on reactive maintenance and manual inspections are increasingly insufficient in resolving the unique challenges presented by NEVs (Gupta et al., 2021). These mainstream techniques can be time-consuming and may not efficiently capture the nuances of system performance, culminating in prolonged vehicle downtime and customer dissatisfaction. As a consequence, there is a pressing need for newer ways and means that use advanced technologies for improved fault prediction and management (Sumsuzoha et al., 2024).

Purpose of the Study

This research project aims to examine the deployment of AI in forecasting and optimizing fault management in NEVs. This study intends to leverage machine learning algorithms with data analytics to provide high reliability and operational efficiency within the US automotive industry with NEVs. The integration of AI into fault management processes may well revolutionize how manufacturers approach vehicle maintenance-from reactive to proactive in a big way. The overall aim is to provide insight concerning how AI has been or may be usefully employed in the prediction of faults before they occur so that timely interventions may be enacted to improve vehicle performance and customer satisfaction. This paper will help add to the growing literature on how AI is going to shape the future of transportation by addressing some limitations in traditional fault detection methods.

Research Questions

RQ1: How can AI models effectively predict faults in NEVs?

- This research question aims to ascertain the methodologies and algorithms that can be employed to analyze vehicle data, identify patterns, and predict potential faults before they manifest.

RQ2: What optimization techniques can be applied to improve fault management in NEVs?

- This research question centers on pinpointing strategies that can elevate the efficiency of fault detection, diagnosis, and resolution processes, ultimately reducing downtime and maintenance costs.

RQ3: How will these AI-driven strategies impact the reliability and efficiency of NEVs in the US market?

- This inquiry seeks to explore the broader implications of executing AI in fault management, comprising its potential to enhance customer satisfaction, reduce operational costs, and enhance the overall reputation of NEVs.

Literature Review

New Energy Vehicles (NEVs)

As per Shrimal (2024), NEV involves all kinds of vehicles that are powered by alternative sources of energy to reduce carbon emissions and decrease dependency on fossil fuels. This category, in the main, comprises electric vehicles, hybrid electric vehicles, and plug-in hybrid electric vehicles. NEVs are positioned for critical roles in the transition of the global automotive industry towards sustainable transportation. They offer the possibility of being a non-polluting alternative not only to traditional internal combustion engine vehicles but also to respond to related governmental policies that deal with climate change and energy independence.

As more countries of the world, like the United States, determined to cut greenhouse gas emissions, the role of NEVs has emerged as increasingly important. All these are empowered by the advancement of technology in batteries, the building of charging infrastructure, and also different forms of government incentives that favor cleaner vehicles (Shern et al., 2024). Thus, there is a proportionate growth in the manufacturers' engagement in research and development into performance improvement, extension of the range of travel of electric vehicles, and price accessibility that works in favor of making these NEVs firm alternatives in transportation (Ukoba et al., 2024).

Sathya et al. (2024), posits that during the last decade, the improvement of NEV technologies has considerably improved, driven by increasingly high demand from consumers and growing regulatory pressures. Key recent trends in this area have been improvements in battery technology the development of solid-state batteries with higher energy density and faster charge times and improvements in electric drivetrains and regenerative braking, allowing efficiency and performance to increase further in NEVs.

The integration of smart technologies like IoT connectivity and ADAS has already brought or promised a sea change in the landscape of NEVs. The key benefits of these innovations, improving vehicle safety and user experience, also facilitate the collection of real-time data that will provide insights into predictive maintenance and fault management. As NEVs continue to evolve, this makes reliability and performance the biggest challenge for the auto sector. Consumer trust and satisfaction rely heavily on this (Muthukumar et al., 2024).

Faults Prediction in Automobiles

Traditional Methods of Fault Prediction and Their Limitations

Rehan (2024), indicated that traditional methods of fault prediction have included scheduled maintenance, manual inspections, and diagnostic tools. Most of these approaches are reactive, with faults usually detected after their occurrence, thus involving costly repairs that may result in vehicle downtime. Some of the techniques used include fault tree analysis and failure mode and effects analysis, but most of them lack dynamic capabilities to address modern NEV complexities.

One major limitation of such techniques is that they would draw upon historical data or even expert knowledge, both of which may not cover all the fast-evolving technologies and diversified usage patterns of NEVs. Besides, the increasing sophistication of vehicle systems, especially in NEVs, which are packed with state-of-the-art software and hardware components, makes fault prediction particularly challenging. There is, correspondingly, growing recognition of the need for more sophisticated, active approaches to fault prediction with increased capacity to accommodate such complexity in modern vehicles (Noori et al., 2021).

Overview of AI Techniques for Fault Prediction

As per Giri et al. (2024), AI has immense transformational potential in fault prediction within the automotive industry. Machine learning algorithms can analyze a huge amount of data generated by vehicle sensors, and identify patterns and anomalies that may indicate impending faults. Techniques such as supervised learning, unsupervised learning, and reinforcement learning are being employed to enhance predictive accuracy and operational efficiency. The historical maintenance records, real-time sensor data, or external factors like weather conditions form the feeding stream for these AI-driven fault prediction systems. This kind of system inherently gets wiser over time with increased exposure and will leap over an unparalleled edge for predictive maintenance scheduling. These, in turn, would promote enabling strategies from reactive to proactive. The impact would be noticed: radical reductions in operational costs, enhancements in vehicle reliability, and, of course, heightened customer satisfaction (Franki et al., 2023).

Optimization Techniques for Fault Management

Garikapati et al. (2024), contended that thhe general fault management of the automotive industry today is usually based on routine checks, diagnostic tests, and maintenance schedules provided by manufacturers. While these techniques may work well, they are usually burdened with inefficiencies in identifying the root cause of the fault and delays in addressing the maintenance needs. Furthermore, these traditional practices may not leverage the full potential of data generated by NEVs.

The biggest challenges in vehicle fault management include the difficulty arising in the interactions involved during diagnosis. A single failure or malfunction may involve the interrelationship of several components interacting; this complicates its diagnosis. This has proved to be an enormous challenge in NEV because there could be close interdependency of software with the hardware and potential cascading outcomes from the integration of many technologies. There is, consequently, a great demand for new optimization techniques to be developed to work on the enhancement of the processes of fault management in vehicles and, generally speaking, improve performance (Ajao, 2024).

AI-Driven Optimization Techniques for Fault Management

AI-driven optimization techniques are therefore promising solutions to the challenges posed by fault management. Analytics of data and machine learning could improve diagnostic processes concerning efficiency and maintenance scheduling. For instance, predictive maintenance algorithms can analyze real-time data to predict possible failures and enable timely intervention that minimizes downtimes. Artificial intelligence makes for the development of an intelligent system able to schedule maintenance based on fault criticality and priority as affecting vehicle performance (Ajao, 2024). It ensures better resource utilization while leaving customers satisfied that issues, especially those critical, do not go unsolved over considerable time. In addition to this, AI optimizations can learn and grow like vehicles by identifying patterns in past events. This forms knowledge necessary for the improvement of their predictions over time.

AI Applications in NEVs

Retrospectively, AI application in NEVs extends beyond mere fault prediction to include several aspects of vehicle performance optimization. AI technologies are being applied in the vehicle's systems for energy management, improvement of driving dynamics, and optimization of charging strategies. For example, AI algorithms can analyze driving patterns and optimize battery use to expand the range of electric vehicles (Dubois, 2022).

Another very vital application of this idea is in fault prediction, where AI-driven diagnostic tools introduced or in the process by some manufacturers apply processes for machine learning on sensor data for fault prediction that are most likely to happen when that fault has not set in (Gupta et al. 2021). Such applications extend reliability for NEVs in ways manufacturers will push into more tailored solutions while improving overall satisfaction since these solutions can offer higher-value services to their clientele.

A growing body of research evidence underlines the successes of AI implementation in NEVs. For instance, a case study on one of the leading electric vehicle manufacturers showed that it utilized machine learning algorithms to predict instances of battery failure and, therefore, was able to reduce warranty claims by a fairly significant percentage, restoring customer trust in the brand. Other studies have reviewed the use of AI inefficient charging station networks and have proposed that smart routing algorithms cut down waiting times and improve user experience (Bukya et al., 2024). Previous research findings reveal that fault management AI applications can have great cost savings for manufacturers in being able to predict faults with high accuracy and optimizing the maintenance schedule, hence minimizing operational interruptions and improving lifecycle management for vehicles. With the continuous evolution of the field, much more research is needed that can further reveal how artificial intelligence can improve reliability and efficiency in NEVs to cater to the demands of grown consumers in the competitive market (Mamatha et al., 2024).

Data Collection and Preprocessing

Data Sources

The dataset for the present study was accessed from accredited automotive manufacturing companies. The dataset was designed to predict the faults and maintenance optimization at NEVs. It covered simulated real vehicle data, such as sensor readings, environmental factors, driving patterns, and maintenance logs that are needed to understand performance, diagnose faults, and optimize a vehicle's maintenance schedule. The dataset was from many different vehicle models and therefore includes both real-time operation data and historical maintenance records (Ziya, 2024). The dataset was designed to assist machine learning algorithms in fault prediction, fault diagnosis, and maintenance optimization, therefore enabling the development of more advanced solutions to improve the performance and longevity of electric vehicles.

Data Preprocessing

The Python code snippet described the step-by-step data pre-processing for any machine-learning task. It first began with the splitting of data into features and target variables using pandas, ensuring specifically that 'fault_type' was isolated as a target. The subsequent step identifies categorical and numerical columns, using a predefined list of categorical variables, and leveraging the Pipeline class from sci-kit-learn to streamline the scaling of numeric features by standardization. Thirdly, the preprocessing of categorical data was an important step, whereby OneHotEncoder was used to transform the categorical variables into a readable format for machine learning models. Fourthly, the preprocessed features are then joined together into one data frame to ensure both types of data are integrated cohesively. Fifth, the data was split into training and testing sets using train_test_split; this helps evaluate model performance. Finally, both the training and test sets have been transformed into a form that is more consistent with other analyses, having interpretability due to the clear mapping of the original indices. This ultimately has structured the dataset in such a way for modeling that it is more time-efficient to extract meaningful information from it.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis is a pivotal step in the analysis of data, whereby the main characteristics of the dataset are summarized and visualized to uncover patterns, anomalies, and relationships among variables. This study relies on EDA to gain insight into the underlying data structure concerning New Energy vehicle fault occurrence and factors contributing to it. EDA develops predictive models for fault management by pointing out trends and their correlations through descriptive statistics, visualizations, and correlation analyses. By highlighting potential issues inside the data, EDA facilitates informed decision-making, guiding the choice of appropriate methods for analysis and ensuring subsequent modeling efforts are based upon a proper understanding of what exists in the data landscape. Therefore, EDA lays the necessary ground for the ensuing improvement in the reliability and operational efficiency of NEVs.

Distribution of Fault Types

The computed Python code snippet created a distribution of fault types in a given dataset. It first imported important libraries such as matplotlib.pyplot and seaborn to generate a variety of plots. Using Seaborn's counterplot, the code generates a bar chart representing the frequency of each fault type in the dataset. The plot has a title, labels, and a color palette for better readability, customized. Finally, it displayed the plot with the plot. Show () as showcased below:

Output:

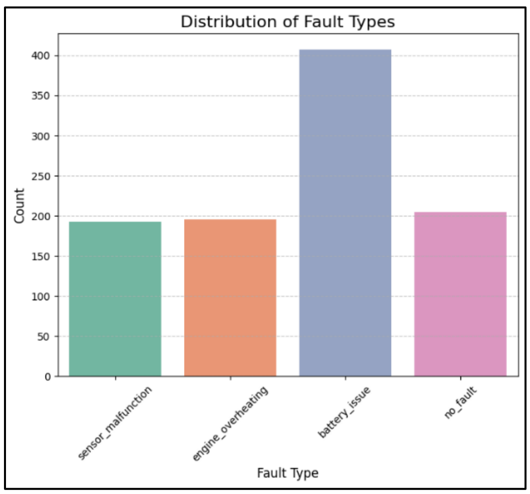

Figure 1: Distribution of Fault Types

The bar chart above represents the distribution of the types of different faults in the dataset. The most frequent is "battery_issue," which happens nearly 400 times. Then, "sensor_malfunction" and "engine_overheating" each happened around 200 times. More interesting is that even "no_fault" is a considerable category because it has happened around 200, meaning quite a fair number of the instances are those that are not considered faulty. A good view comes from that on types of issues encountered versus their respective frequencies.

Battery Voltage Across Fault Types

The computed Python code was designed to create a box plot visualization using the Seaborn library. In essence, it explored how battery voltage varies across different fault kinds in some datasets. That is, the code sets up a figure with the inch dimensions of 10 x6 inches, creating a box plot by mapping the, identifying its source as 'data', x as 'fault-type', and y as 'batter_voltage'. It also sets the color palette to "pastel" for an appealing presentation. The title of the plot is set to "Battery Voltage Across Fault Types," and the x- and y-axis labels are labeled appropriately with font sizes. Finally, the line plot.sticks (rotation=45) rotate the x-axis labels by 45 degrees to improve readability if the fault type names are long. The plot is then displayed using plt.show() as displayed below:

Output:

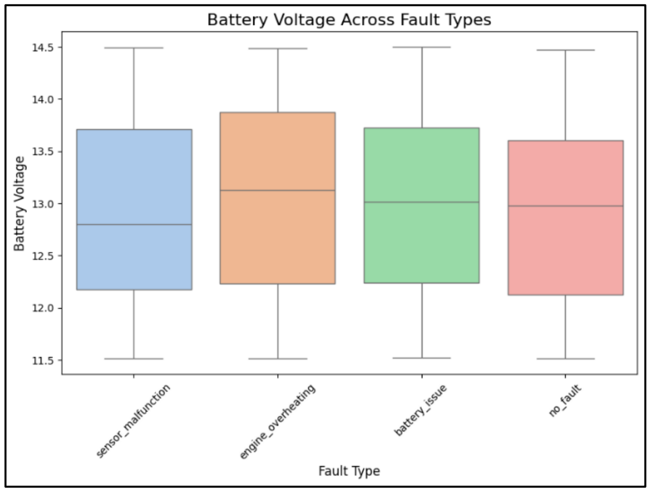

Figure 2: Displays Battery Voltage Across Fault Types

The boxplot depicts the distribution of a battery voltage for various kinds of faults. We observe that the median of a battery voltage for all the types of faults is almost equal to 13 volts with the difference in dispersion. Therefore, the "sensor_malfunction" group has the lowest range, which indicates small variability in the battery voltage for these instances. In contrast, it can be seen that the "no_fault" group presents the biggest range, and thus is expected to bear the widest spread concerning voltage battery value when no kinds of faults were recorded. The findings would hence insinuate that while, perhaps, the battery voltage may not be a primary pointer in the type of fault determination, it could hitherto be employed to supplement ancillary insights into the system health and performance of such a battery.

Road Condition Distribution

The Python code script was designed to make a counterplot to show the distribution of road conditions in a given dataset. As a consequence, it created a figure with 8x6 dimensions using plt-figure (). Then, it uses the SNS. Counterplot () function from the Seaborn library, creating the counterplot with the dataset as data, the x-axis set to the road_condition column, and a color palette. The plot then got the title "Road Condition Distribution" in a certain font size. Then, label the x-axis as "Road Condition" and the y-axis as "Count", with their corresponding specified font sizes. It also included the x-axis labels to be rotated 45 degrees for readability. It finally added a grid in the y-axis with the given line style and transparency and plt.show () was used to display this plot as showcased below:

Output:

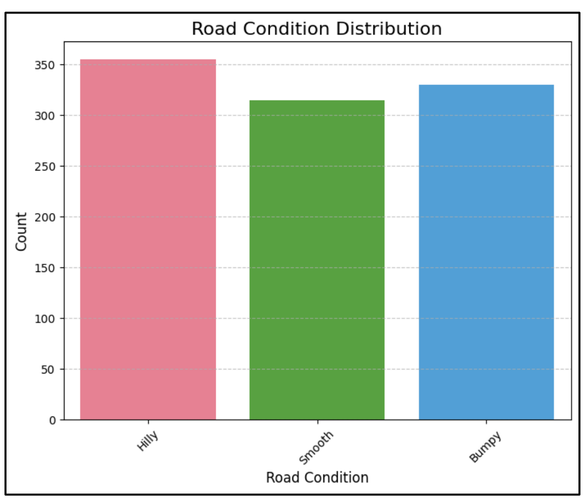

Figure 3: Exhibits Road Condition Distribution

The bar chart below depicts the distribution of road conditions available within the dataset. The frequent condition is "Hilly," with about 350 instances. Then there are "Smooth" roads, which have been found around 300 times. Finally, "Bumpy" roads are recorded to have occurred around 325 times. This plot creates great insights into the prevalence of different road conditions, and it shows that this dataset has more hilly and bumpy roads rather than smooth roads.

Pair Plot of Selected Features

The code snippet in Python, using the Seaborn library, was meant for creating a pair plot and studying relationships that may exist among a selected set of features in some data, and how these vary with different types of fault. Specifically, this selects 'battery voltage', 'engine temperature', 'speed', 'fuel efficiency', and 'fault type' from the data. Then, sns. Pair plot () had to be executed to create the pair plot; select the features for hue to paint data points with different colors regarding their fault type, while the palette is chosen to 'tab10' since this looks nice, meaning it offers visually very nicely distinguishable colors; and set diag_kind='kde', since kernel density estimation plots look nice as diagonal elements within this kind of pair plot. Finally, the plot is given a title "Pairplot of Selected Features" with appropriate font size and positioning, and the plot is displayed using plt.show().

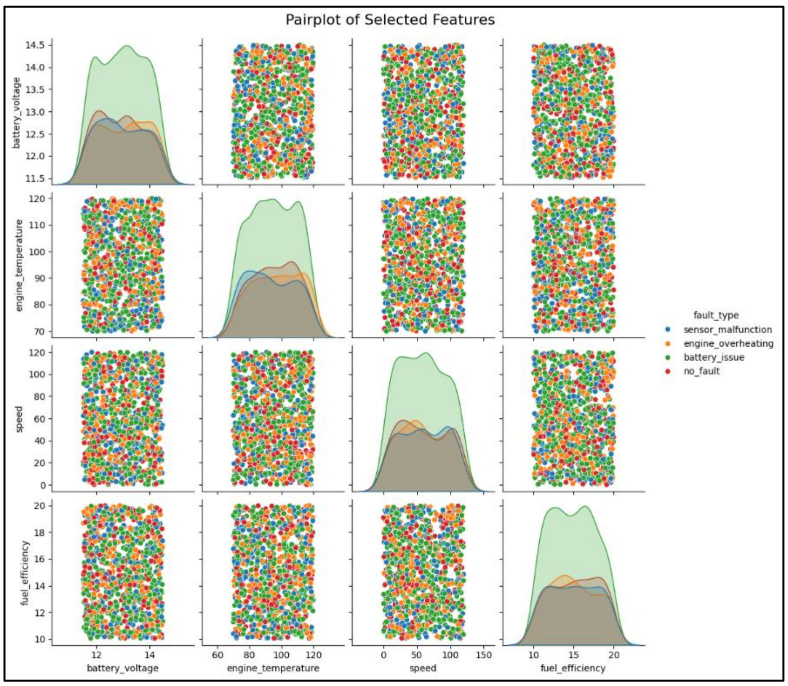

Figure 4: Portrays Pair Plot of Selected Features

The pair plot above provides a full overview of the relationships of selected features ('battery_voltage', 'engine_temperature', 'speed', and 'fuel_efficiency') with each other and the relation to different fault types. We can see a few interesting patterns. There is a small positive correlation between battery voltage and engine temperature; we can see that from the upward trend in the scatter plot. Furthermore, the trend on fuel efficiency is different from that for different fault types in which some of them have higher density within the low fuel-efficiency region. The plot above, a pair-plot, also depicts other important relationships that might exist amongst other pairs of features showing avenues for further in-depth analysis into how these various variables could interact and correlate to determine fault occurrence in this model.

The computed Python code snippet was intended for creating a scatter plot with the Seaborn library. The aim was to explore the relationship between Service Frequency and Repair Cost, color-coding the points according to Fault Type. This will create a figure with size 10x6 and make the scatter plot using sns.scatterplot() on data with the x-axis as service_frequency y-axis as repair_cost and hue as fault_type. The color palette used here is 'viridis' and alpha is 0.8. Set the title to "Service Frequency vs. Repair Cost" and give the appropriate font size. Label the x and y axes, giving the appropriate font size for each. Finally, display the plot using plt.show().

Output:

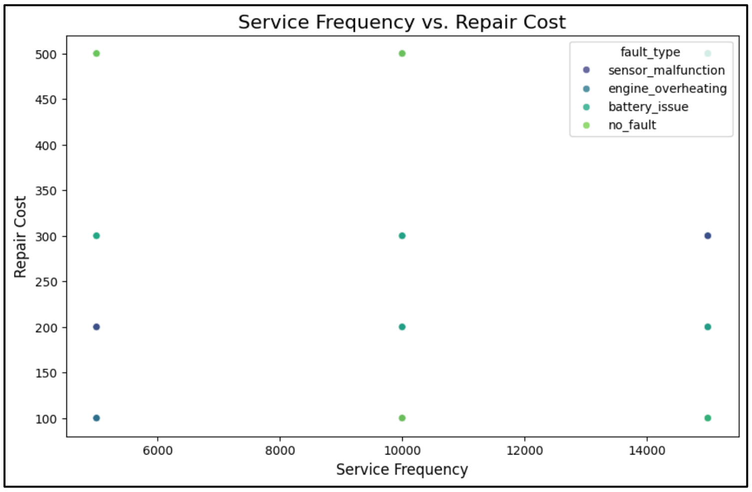

Figure 5: Visualizes Service Frequency vs. Repair Cost

The scatter plot shows the relationship between service frequency and repair cost, where each point is color-coded according to its corresponding fault type. We can notice some interesting patterns. In particular, there seems to be a positive trend between service frequency and repair cost-for example, as service frequency increases, so do repair costs. However, it is also not a perfectly linear relationship in that some points lie off of this general trend. Again, the scatter plot displays different fault types exhibiting varying repair costs at different service frequencies. For instance, it appears that "sensor malfunction" cases always have lower repair costs independent of service frequency. This visualization provides much-needed insights into the financial implications of service frequency and its relation to fault types.

The code in Python attempted to build a scatter plot, using the Seaborn library to study the correlation between ambient temperature and humidity, with colors mapped to the road conditions. The code created a figure with 10-inch width and 6-inch height using sns.scatterplot() plot settings with arguments: dataset='data', x-axis variable='ambient-temperature', y-variable= 'humidity' but coloring (Hue) to vary by 'road-condition ', palette='cool', is visually appealing. Alpha- 0.8 for some transparency towards overplotting visibility;. This sets the title for this plot to "Ambient Temperature vs. Humidity by Road Condition" along with a suitable font size. Labels are added on x and y-axis correspondingly, using suitable font size. Lastly, the plot was exhibited and viewed with plt.show() as presented below:

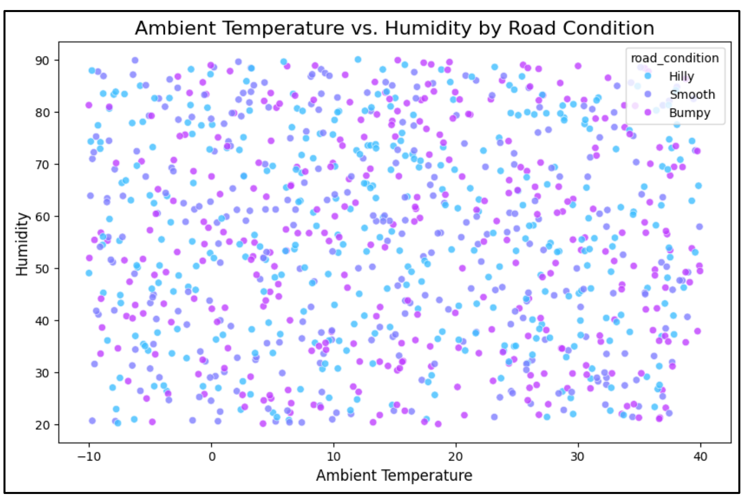

Figure 6: Showcases Ambient Temperature vs. Humidity by Road Condition

The scatter plot maps the variation of ambient temperature versus humidity, where color-coding was done based on the condition of the road, hilly, smooth, or bumpy. The plot has shown a big variation both in ambient temperature and humidity, but no specific clustering or patterns have emerged in this plot. The data points representing the entire three road conditions are sprinkled all over the plot; therefore, there is no strongly identified correlation between ambient temperature-humidity and road condition variations from this dataset. This would now probably suggest that other more controlling factors like road construction and maintenance may play the role of determining the road conditions in such a case.

Methodology

Feature Engineering and Selection

Feature engineering and selection are among the most important steps in the machine learning workflow that mainly aims at improving performance for predictive models. Feature engineering is a process to transform raw data into informative features that can better capture meaningful patterns related to the problem at hand. These techniques for feature extraction and engineering of relevant features from the dataset may include the creation of interaction terms, polynomial features, and aggregating data over time or categories. For example, in fault prediction from the NEVs' perspective, one may feature-engineer any number of average temperatures a battery has been running for, frequency of maintenance events, or even cross-terms of vehicle speed and battery efficiency. The nature of these features could bring in a different level of context that raw data alone would not be able to do, thus enabling the model to learn complex relationships present in the data.

Of equal importance are criteria on the selection of most predictive features, as not all features contribute positively towards model performance. A standard method for feature selection considers some techniques like correlation analysis; features are screened with regards to their statistical dependence with the target variable. Herein, fault types will act as target variables. Furthermore, other techniques such as Recursive Feature Elimination and feature importance from tree-based models can also be used to determine which features bear the most influence on the predictions. This will help balance model complexity with interpretability, making sure that not only does the choice of features enhance predictive power but also remains tractable and understandable. This careful curation of features is essential for building robust models that can generalize well to unseen data.

Model Selection

Fundamentally, fault prediction depends on the selection of appropriate machine learning and deep learning models. Different algorithms were selected, such as Random Forest Classifier, Gradient Boosting Classifier, and Logistic Regression-all with different advantages, depending on the characteristics of the dataset and the complexity of the problem. The Random Forest Classifier was appropriate for high-dimensional datasets, which can handle numerical and categorical features efficiently. Its ensemble approach avoids overfitting and hence can be an excellent choice to predict faults in NEVs where there may be a lot of interacting features.

On the other hand, the Gradient Boosting Classifier is known to perform well in boosting the performance of weak learners with a high degree of accuracy and robustness against overfitting. Due to its iterative in nature, it can hone in on fixing errors made by other models in the sequence; this really helps when modeling complex relationships. Logistic Regression has much in common with those, though simpler, as a good baseline model of binary classification tasks. Further, due to its interpretability, understanding how features contribute to making predictions will provide valuable insight into automotive applications, fault mechanism insight that leads to actionable insight.

The choice of model was justified on the premise of characteristics of the data: dimensionality, feature types, and nature of the target variable. Also, the goal of the prediction task guides model selection, for example, whether to maximize accuracy to minimize false positives or to ensure interpretability. In this work, several models are combined that leverage strengths to enable a more complete view of the landscape of fault prediction.

Model Development and Evaluation

Model development involves the training and testing of the selected models using the collected data. First, the dataset is divided into a training set and a test set to make sure that the model performance can be evaluated on unseen data, thus providing a realistic estimate of the performance of the models. The models learn from the features and their corresponding target values in training to optimize their internal parameters and reduce prediction errors. This phase is very important to develop a model which generalizes well on new data.

Cross-validation techniques, including k-fold cross-validation made sure that the performances of these models are robust. It worked by dividing the training dataset into k subsets and training the model on k-1 subsets, then validating it on the remaining subset. This protocol was repeated k times, each time using one subset for validation. Cross-validation helps mitigate the associated risks of overfitting and provides a more real-world estimate of the model's predictive performance by evaluating the model across different subsets.

Other important areas of model development included tuning the hyperparameters, which are crucial to optimizing the performance of a model. The hyperparameters are configurations set from the outside before training, like the number of trees in Random Forest and the learning rate in Gradient Boosting. Approaches like Grid Search or Random Search were used, which were systematically explored within a range of hyperparameters to determine the best mix of them that yields superior metrics. This tuning ensures that the model lives up to its full potential and caters well to the nuances within the data.

Performance evaluation of the model was done with several metrics, most notably precision, recall, F1-score, and ROC-AUC. Accuracy gives a general measure of the model's correctness, but can be misleading for imbalanced datasets. Precision and recall provide insight into how well the model is able to identify correctly the positive cases, those with faults, while avoiding false positives - something particularly relevant in scenarios involving fault prediction. F1-score yields a harmonic mean of precision and recall into one single metric balancing both. Lastly, ROC-AUC examines the performance of a model across various threshold settings for distinguishing classes and offers a more robust performance metric, especially when working on binary classification problems. The use of such metrics would mean that the study develops models that are accurate and, at the same time, reliable and efficient in the prediction of New Energy Vehicles faults.

Results and Analysis

Model Performance

a) Logistic Regression

This snippet of Python code performed a logistic regression modeling. It started by importing the necessary libraries: Logistic Regression and performance metrics-classification report, accuracy score. Next, it created a pipeline for smoothly going through the pre-processing stage, assumed to be defined elsewhere, and the logistic regression classifier itself. Then, the model was trained with the training data and used to predict the test data. The model was finally evaluated by printing out an accuracy score and a classification report showing its precision, recall, and F1 score on both classes as showcased below:

Output:

Table 1: Displays Performance of Logistic Regressions

```
Performance of Logistic Regression:
Accuracy: 0.3550
Classification Report:
                    precision    recall  f1-score   support

     battery_issue       0.37      0.89      0.52        76
 engine_overheating       0.27      0.09      0.13        34
          no_fault       0.00      0.00      0.00        48
sensor_malfunction       0.00      0.00      0.00        42

          accuracy                           0.35       200
         macro avg       0.16      0.25      0.16       200
      weighted avg       0.19      0.35      0.22       200
```

From the classification report, we observe that the Logistic Regression model achieves an accuracy of 0.3550, which is moderately adequate. Also, within-class metrics indicate that for "battery issue" the model has a high recall of 0.89 but low precision of 0.37, showing many false positives; whereas for "engine overheating", it has a low recall of 0.09 and low precision of 0.27, showing poor performance in identifying this class. The model works really bad for "no fault" and "sensor malfunction" classes, obtaining zero precision, recall, and F1-score, which shows it is not able to correctly classify these instances. Also, the model is performing very low according to the macro and weighted average metrics. These scores denote that the model needs further tuning to classify all fault types well.

b) Random Forest Modelling

The Random Forest Classifier model was equally implemented in Python. The script first imported the necessary Random Forest Classifier from the sklearn. ensemble library. It then instantiates the pipeline for processing, made up of a preprocessing step assumed to be defined somewhere and, finally, the Random Forest classifier itself. Thereafter, the training data, X-train, y-train, was used in training the model while making its predictions on the test set X_test. Finally, the code evaluated the performance of the model by calculating and printing the accuracy score and a classification report, which provided insight into the model's precision, recall, and F1-score for each class as displayed below:

Output:

Table 2: Portrays Performance of Random Forest Classifier

```
Performance of Random Forest Classifier:
Accuracy: 0.3850
Classification Report:
                    precision    recall  f1-score   support

     battery_issue       0.40      0.95      0.56        76
engine_overheating       0.43      0.09      0.15        34
          no_fault       0.40      0.04      0.08        48
sensor_malfunction       0.00      0.00      0.00        42

          accuracy                           0.39       200
         macro avg       0.31      0.27      0.19       200
      weighted avg       0.32      0.39      0.26       200
```

The classification report for the Random Forest algorithm achieved an overall accuracy of 0.3850, indicating a slightly better performance than the Logistic Regression model. Once again, even though highly effective in yielding high recalls, as seen in "battery_issue" with 0.95, the highest in precision is only an average of 0.40. Similar to LR, it struggles with three classes: "engine overheating," "no fault", and "sensor malfunction," managing zero each in precision, recall and F1-score for latter two. The above metrics are the perfect reflection of the model's overall problem, as evidenced by the macro and weighted average. These results show that the Random Forests model, though improved rather compared to Logistic Regression, still needs further optimization to give a good classification for all forms of fault types.

c) Gradient Boosting Classifier

The code snippet in Python implemented the Gradient Boosting Classifier model. It commenced by importing the library necessary for the Gradient Boosting Classifier from sklearn. ensemble, creates a pipeline for easy flow, and consists of preprocessing-preprocessing that is presumably defined elsewhere-and a Gradient Boosting classifier. Finally, the model was trained on the training data, X_train, y_train, and made its prediction on the test data, X_test. Finally, the code evaluated this model by computing and printing the accuracy score and also a classification report, both of which contain the results of the model's precision, recall score, and F1-score for classes:

Output:

Table 3: Illustrates Gradient Boosting Classifier

```
Performance of Gradient Boosting Classifier:
Accuracy: 0.3650
Classification Report:
                    precision    recall  f1-score   support

     battery_issue       0.40      0.80      0.54        76
engine_overheating       0.36      0.26      0.31        34
          no_fault       0.38      0.06      0.11        48
sensor_malfunction       0.00      0.00      0.00        42

          accuracy                           0.36       200
         macro avg       0.28      0.28      0.24       200
      weighted avg       0.30      0.36      0.28       200
```

The Gradient Boosting Classifier has an accuracy of 0.3650-outperforming the Logistic Regression model but still falling behind the Random Forest Classifier. Although it gives very good recall for "battery_issue" at 0.80, it has very low precision at 0.40. It follows the trend of earlier models in having low performance on "engine_overheating," "no_fault," and "sensor_malfunction" classes while managing zero precision, recall, and F1-score for the latter two classes. The macro and weighted average metrics reflect the general limitations of the model. These results clearly indicate that the Gradient Boosting model, while promising and performing better for some classes than Logistic Regression, needs further optimization in order to classify all fault types effectively.

Comparison of All Models

The code in Python compared several performances of machine learning techniques, namely Logistic Regression-Random Forest-Gradient Boosting. The script imported required libraries, among those: matplotlib. pyplot, a seaborn, and accuracy _score. Then, estimated the accuracy for each of, them by using the accuracy function. After that, store estimated accuracies in a dictionary and convert it over pandas Data Frame for easier, prettier visualization. Finally, the code printed out the performance table, using seaborn for a bar plot to visually compare the accuracy of the three models. Finally, the plot was labeled accordingly with labels, title, and other formatting for good readability.

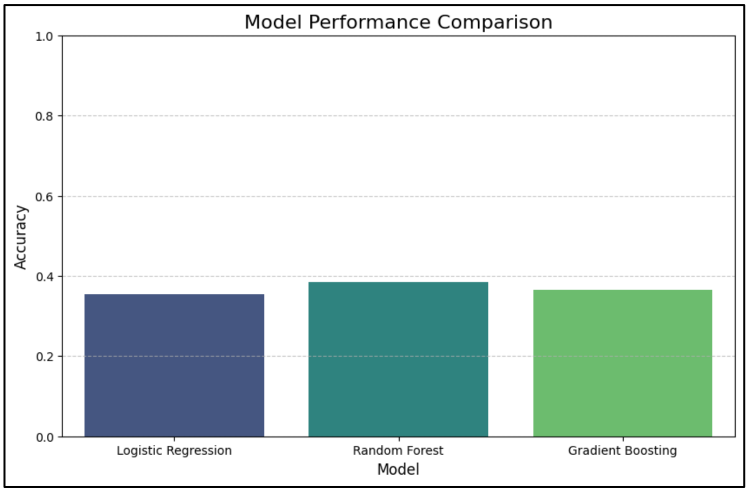

Figure 7: Visualizes Model Performance Comparison

The bar chart showcases visual comparisons of accuracy scores that Logistic Regression, Random Forest, and Gradient Boosting were able to generate in three machine-learning models. The results demonstrate that the Random Forest model attained the highest accuracy, followed closely by Gradient Boosting. Logistic Regression had the lowest accuracy among the three models. These findings suggest that among those based on pure accuracy, the Random Forest would be the most promising option for the task at hand. However, other metrics, such as precision and recall, including the F1-score, are also useful to gain a better overall understanding of the results of each model.

Feature Importance Analysis

The most important role of feature analysis in any machine learning model is understanding the root causes of the predictions, especially for New Energy Vehicle fault prediction. Models such as Random Forest and Gradient Boosting have some intrinsic mechanisms to estimate feature importance that may guide the engineers and decision-makers on which are the most important variables related to faults. For example, with Random Forest, it gives you the contribution of the importance of the feature used with Gini impurity or mean decrease in accuracy toward the model's predictability. These importance scores over all the trees sum to show you which features generally always come as important over several random subsets. This allows stakeholders to concentrate on the most influencing variables, enhancing the interpretability of the model and thus targeted interventions in strategies of fault management.

On the other hand, feature importance in Gradient Boosting is calculated with respect to the reduction of the loss functions upon the addition of a feature in the model. The idea behind this approach is to provide a nuanced view: it considers how features interact with each other and their cumulative impact on the performance of the model in question. By analyzing feature importance scores derived from both models, one can identify the key contributors to fault prediction. Instances of top contributors may be battery temperature, charging frequency, vehicle speed, and historical maintenance records.

Predictive Insights

Interpretation of the model's predictions is important in fault management strategies because it converts raw predictive outputs to actionable insights. Studying the predictions of models like Random Forest and Gradient Boosting gives a better understanding to stakeholders on potential faults and causes. For instance, if the model predicts a high probability of battery failure in a given condition, such as high temperatures combined with long charging time, the insight can be used to inform preventive measures. In this regard, maintenance teams can then focus their inspection or cooling solutions to decrease the risk of battery degradation and improve vehicle reliability.

Moreover, the insights from predictive models may assist in communicating better with customers. With transparent explanations of why something is predicted to fail, the manufacturers can help reassure the users. For instance, if a model comes back with a high likelihood of a pending fault based on driving patterns or battery usage type, the manufacturer can provide certain suggestions to the vehicle's owner about how to regulate his or her usage in particular.

Furthermore, predictive insight also allows the curation of dynamic maintenance schedules responsive to current data and not necessarily bound by rigid time frames. By integrating all these insights into their systems, companies can do better fleet management to ensure necessary maintenance on vehicles is correctly timed, when the vehicles are most likely to develop faults. From reactive to proactive, the maintenance itself has shifted, thereby not only indirectly contributing to the improvement in operational efficiency but also the overall sustainability of NEVs due to a decrease in downtime and full vehicle availability. Interpretation of model predictions is important for translation of insights drawn from data-driven information into practical ways of improving performance through better fault management of New Energy Vehicles.

Implementation Strategy

Integration into NEV Operations

In such new-energy vehicle-integrated fault-predictive models, AI needs to be woven into an integral unit in the form of a system and carefully planned for easy integration with the best results. The first step in that direction would be an appraisal of the existing data infrastructure regarding NEV systems concerning types, sources, vehicle performance data, sensor data, and maintenance records. It therefore requires developing a strong data pipeline that would ensure the continuous gathering and storage of the most relevant real-time data. This sometimes involves the deployment of various IoT devices and sensors able to capture detailed operational data in great detail from different components of vehicles.

After the data architecture is in place, the next step is to implement the trained fault prediction models into the NEV's onboard systems. That would require close cooperation between data scientists and automotive engineers to ensure the models are harmonious with vehicle architecture and will run seamlessly within the given computational constraints. Once implemented, the models should be interfaced with existing vehicle diagnostic tools to enable real-time monitoring and analysis. A user-friendly interface should be devised to provide presents predictive insights to maintenance personnel and vehicle operators, who can then take quick action based on the model's recommendations. Finally, continuous monitoring and updating of the models are important to ensure their accuracy and relevance over time, as new data and evolving technologies may necessitate retraining or fine-tuning the algorithms.

Such a proactive approach will result in better customer satisfaction, while at the same time lowering warranty claims and maintenance costs by the manufacturers.

Such understanding of the relationship will be useful not only for model performance enhancement but also for the design of maintenance protocols and engineering solutions to tackle the root causes of faults. By giving priority to such key factors, manufacturers can optimize resources and improve the reliability and efficiency of NEVs.

Scalability and Flexibility

The scalability and flexibility of the AI-driven fault prediction models should be realized with their successful deployment across different types of NEVs and operational scales. Scalability in this context means that the models can handle increased volume in data and complexity while the fleet grows or new models are introduced. This may be realized using cloud-based platforms able to store large datasets and provide computation resources for real-time processing. More the use of in-service vehicles, larger the data set on which the models could be retrained again at improving their predictions by accommodating different driving conditions and modes of use.

Flexibility is equally essential, as NEVs come in distinct forms—ranging from completely electric vehicles to hybrid algorithms—and each type may present unique operational challenges. Fault prediction models have to be flexible, which means their design must include a modular architecture that easily adapts to changes for different specifications of vehicles and their respective features. Algorithms should be extensible so that they can easily cover new types of data, like newly developed sensor technologies, and other sources reflecting changes in users' behavior. Scalability and flexibility will allow AI-driven fault prediction systems from manufacturing to be relevant and effective against the changing landscapes of NEVs.

Business Impact Analysis

Estimating the potential business impact of the implementation of AI-driven fault prediction and optimization includes an analysis of how these technologies can enhance operational efficiencies, reduce costs, and enhance customer satisfaction. One of the biggest advantages of all is to reduce maintenance costs by proactive detection of faults. By identifying upcoming issues before they develop into major failures, manufacturers can reduce not only the frequency but also the severity of repairs and, simply put, reduce any claims on warranty or service costs to as low as possible. Better reliability in vehicles also leads to better customer satisfaction, which then affects brand loyalty and repeat purchases.

A full cost-benefit analysis of the implementation would include the initial investment in AI-driven systems and long-term savings. Initial costs could include software development, integration, and training personnel to use the new systems effectively. On the other hand, these upfront costs can be offset by the long-term savings from fewer breakdowns and reduced downtime, with added value through optimized maintenance schedules. In addition, this capability to provide better predictive insights brings more revenue streams, as it can charge subscription-based services for real-time monitoring or even advanced analytics. Ultimately, the quantifying of these benefits through appropriate implementation will go a long way in highlighting the substantive returns on investments possible with AI-driven fault prediction and optimization in NEV operations.

Discussion

Implication for the US Automobile Industry

AI-driven fault prediction models brought into play would greatly raise the level of impact that can be caused to the automotive industry in the US concerning the enhancement of NEV reliability and efficiency. It is expected that the sophisticated algorithms in fault prediction by the manufacturing industry will help avoid random breakdowns, if not great reduction, that could reduce the time the vehicles can put in and customer satisfaction. Such a proactive approach will not only limit repair costs but also lead to increased trust between consumers and manufacturers, as customers start to demand reliable vehicles in the competitive automotive market. The ability to predict will also further enable the manufacturing industry to improve its supply chain and maintain the right inventories of vital parts required by customers while minimizing operational bottlenecks.

To efficiently incorporate predictive algorithms into NEV manufacturing and maintenance protocols, several recommendations should be considered. First, automotive manufacturers will need to develop a robust data infrastructure able to collect and analyze data about vehicle performance throughout its production and operational life cycle. This may involve investments in both IoT technologies and data analytics platforms that can process and analyze large volumes of data in real time. A combination for the development of predictive models addressing practical aspects should be approached, involving cross-functional teams of data scientists, automotive engineers, and maintenance staff. The necessary training programs in essential competencies for relevant staff needed for effective interpretation and insight actioning should also be developed. With these strategies in place, the US automotive industry will fully implement AI-driven fault prediction for entry into a more resilient NEV ecosystem.

Challenges and Limitations

While there are substantial benefits from AI-driven fault prediction, several challenges and limitations in the process of optimization of their effectiveness should be overcome. In this regard, a serious ethical issue touches on the use of automotive data for predictive analytics. The privacy and security of that data are of great concern since gathering and processing sensitive information includes driving habits and performance of a vehicle and involves serious concerns of user consent and data misuse. This will call for clear data governance policies by manufacturers to guarantee user privacy with a strategic goal of engendering trust among consumers. This then suggests clarity in what is collected, how it will be used, and wherever possible anonymized, so it will not compromise the identity of the individuals.

Apart from that, some limitations in data quality, model interpretability, and generalizability restrain the use of AI-driven fault prediction models. The path to getting the right predictive models requires high-quality data, yet inaccuracies in predictions result from inconsistencies, gaps, and biases in the data. To overcome this problem, effective collection of data has to be done, accompanied by stringent data cleaning. Despite AI models being very efficient in the output of highly accurate predictions, interpretability is a double-edged sword. For example, complex models often behave like "black boxes," where stakeholders can't understand why certain predictions are made. This lack of interpretability can make people lose faith in the models and thus complicate decision-making. Finally, there is the issue of generalizability: models that have been trained on specific datasets may not generalize well to different vehicles or operating conditions.

Future Research Directions

In the future, the research directions for NEVs powered by artificial intelligence in fault prediction would be directed to several pathways to enhance model accuracy and effectiveness in operations. One favorable opportunity may come from a larger and more diverse set of datasets. As the population of NEVs increases further, so does the quantity of operational data. Further, researchers should focus on data aggregation from various sources like different models of vehicles, varied environmental conditions, and diverse driving behavior. The richness in this data landscape could facilitate richer model training, thus offering finer predictions that incorporate more variables and complexities.

Moreover, the increased potential of real-time data integration and advanced analytics techniques opens more avenues for innovation. The higher the connectivity that characterizes modern vehicle design, the greater the number of real-time monitoring and in-depth analysis of performance data being generated. Additional research is needed to identify the best ways of tapping streaming data and applying appropriate machine learning techniques for realistic predictions to be made enabling runtime scheduling of maintenance and in good time warnings of likely faults. Finally, other enhanced analytics techniques such as transfer learning or ensemble learning should therefore be considered concerning their potential for improving model ability and adaptiveness. These focused areas would keep the researchers and practitioners continuously working toward fine-tuning the AI-driven fault prediction models, thereby increasing the reliability and efficiency of New Energy Vehicles within an evolving automotive ecosystem. The robustness of these models in diverse scenarios would continuously have to be validated and refined.

Conclusion

This research project aimed to examine the deployment of AI in forecasting and optimizing fault management in NEVs. This study intended to leverage machine learning algorithms with data analytics to provide high reliability and operational efficiency within the US automotive industry with NEVs. The dataset for the present study was accessed from accredited automotive manufacturing companies. The dataset was designed to predict the faults and maintenance optimization at NEVs. It covered simulated real vehicle data, such as sensor readings, environmental factors, driving patterns, and maintenance logs that are needed to understand performance, diagnose faults, and optimize a vehicle's maintenance schedule. Different algorithms were selected, such as Random Forest Classifier, Gradient Boosting Classifier, and Logistic Regression-all with different advantages, depending on the characteristics of the dataset and the complexity of the problem. Performance evaluation of the model was done with several metrics, most notably precision, recall, F1-score. The results demonstrated that the Random Forest model attained the highest accuracy, followed closely by Gradient Boosting. AI-driven fault prediction models brought into play would greatly raise the level of impact that can be caused to the automotive industry in the US concerning the enhancement of NEV reliability and efficiency. Interpretation of the model's predictions is important in fault management strategies because it converts raw predictive outputs to actionable insights.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Ahmad, T., Zhu, H., Zhang, D., Tariq, R., Bassam, A., Ullah, F., ... & Alshamrani, S. S. (2022). Energetics Systems and artificial intelligence: Applications of industry 4.0. *Energy Reports*, *8*, 334-361.

[2] Ajao, O. R. (2024). Optimizing Energy Infrastructure with AI Technology: A Literature Review. *Open Journal of Applied Sciences*, *14*(12), 3516-3544.

[3] Bukya, R., Mohan, G. M., & Swamy, M. K. (2024). Artificial Intelligence Role In Optimizing Electric Vehicle Charging Patterns Reduce Costs And Improve Overall Efficiency: A Review. *Journal of Engineering, Management and Information Technology*, *2*(3), 129-138.

[4] Chelliah, P. R., Jayasankar, V., Agerstam, M., Sundaravadivazhagan, B., & Cyriac, R. (2023). The Power of Artificial Intelligence for the Next-Generation Oil and Gas Industry: Envisaging AI-inspired Intelligent Energy Systems and Environments. John Wiley & Sons.

[5] Dubois, M. (2022). AI-Powered Predictive Analytics for Vehicle Maintenance Scheduling. *Australian Journal of Machine Learning Research & Applications*, *2*(2), 278-293.

[6] Franki, V., Majnarić, D., & Višković, A. (2023). A comprehensive review of Artificial Intelligence (AI) companies in the power sector. *Energies*, *16*(3), 1077.

[7] Garikapati, D., & Shetiya, S. S. (2024). Autonomous Vehicles: Evolution of Artificial Intelligence and the Current Industry Landscape. *Big Data and Cognitive Computing*, *8*(4), 42.

[8] Giri, N., Pandit, Y., Damle, T., Nair, A., Panchal, J., Sharma, N., & Deshmukh, H. (2024). A Novel approach to Services and Fault Prediction Diagnostics of Artificial Intelligence based Smart Driver Assistant of Two-Wheeler.

[9] Gupta, S., Amaba, B., McMahon, M., & Gupta, K. (2021, May). The evolution of artificial intelligence in the automotive industry. In *2021 Annual Reliability and Maintainability Symposium (RAMS)* (pp. 1-7). IEEE.

[10] Mamatha, N., Ramesh, H. R., & Santhosha, D. (2024). Coordinated Operation of Electric Vehicle Charging Stations (EVCS) and Distributed Power Generation in Grids Using AI Technology. In *Distributed Energy Resources and Electric Vehicle* (pp. 137-157). CRC Press.

[11] Muthukumar, R., Pratheep, V. G. G., Sultanuddin, S. J., Kunduru, K. R., Kumar, P., & Boopathi, S. (2024). Leveraging Fuel Cell Technology With AI and ML Integration for Next-Generation Vehicles: Empowering Electric Mobility. In *A Sustainable Future with E-Mobility: Concepts, Challenges, and Implementations* (pp. 312-337). IGI Global.

[12] Noori, I., Abolelmagd, Y. M., & Mobarak, W. F. (2023). Artificial Intelligence and the Decarbonization Challenge. In *Artificial Intelligence and Transforming Digital Marketing* (pp. 849-857). Cham: Springer Nature Switzerland.

[13] Rehan, H. (2024). The Future of Electric Vehicles: Navigating the Intersection of AI, Cloud Technology, and Cybersecurity. *Valley International Journal Digital Library*, 1127-1143.

[14] Sathya, D., Saravanan, G., Senthilvadivu, K., Jeniffer, S. B., Thangamani, R., & Kruthika, M. S. (2024). Demand response management using artificial intelligence-empowered electric vehicles. In *Artificial Intelligence-Empowered Modern Electric Vehicles in Smart Grid Systems* (pp. 115-151). Elsevier.

[15] Shern, S. J., Sarker, M. T., Ramasamy, G., Thiagarajah, S. P., Al Farid, F., & Suganthi, S. T. (2024). Artificial Intelligence-Based Electric Vehicle Smart Charging System in Malaysia. World Electric Vehicle Journal, 15(10), 440.

[16] Shill S. K., Chowdhury, M. S. R., Tannier, N. R., Tarafder, M. T. R., Akter, R., Gurung, N., & Sizan, M. H. (2024). Forecasting Electric Vehicle Adoption in the USA Using Machine Learning Models. Journal of Computer Science and Technology Studies, 6(5), 61-74.

[17] Shrimal, H. (2024). *Integration of AI-Powered Vehicles with Smart City Infrastructure to Transform the Future of Automotive World* (No. 2024-28-0028). SAE Technical Paper.

[18] Sumsuzoha, M., Rana, M. S., Islam, M. S., Rahman, M. K., Karmakar, M., Hossain, M. S., & Shawon, R. E. (2024). LEVERAGING MACHINE LEARNING FOR RESOURCE OPTIMIZATION IN USA DATA CENTERS: A FOCUS ON INCOMPLETE DATA AND BUSINESS DEVELOPMENT. The American Journal of Engineering and Technology, 6(12), 119-140.

[19] Ukoba, K., Olatunji, K. O., Adeoye, E., Jen, T. C., & Madyira, D. M. (2024). Optimizing renewable energy systems through artificial intelligence: Review and future prospects. *Energy & Environment*, 0958305X241256293.

[20] Ziya. (2024, November 16). *Fault prediction and optimization in NEVs*. Kaggle. https://www.kaggle.com/datasets/ziya07/fault-prediction-and-optimization-in-nevs