Generalizing Vehicle Manoeuvre Prediction Across Diverse Datasets

Marella VenkataRao[1],Dodda Venkata Reddy[2],Dr.S.Siva Nageswara Rao[3],
Ameen Ul Hassan khan[4],Golla Avinash[4],Pendela Chenchu Koushik[4],
[1]Associate Professor, [2]Assistant Professor, [3]professor, Department of Computer Science and Engineering,
Narasaraopeta Engineering College(Autonomous)
Narasaraopet, Palanadu, Andhra Pradesh, India
Email:[1]venkatmarella670@gmail.com,[2]doddavenkatareddy@gmail.com,
[4]Email:ameenkhan5256@gmail.com, avigolla5@gmail.com, chenchukoushik@gmail.com

***Abstract*—Due to the complex nature of automotive components and sensor data, predictive maintenance is essential to ensure the reliability and safety of the vehicle. This work introduces a new predictive algorithm for automotive engine health, designed as it will provide higher accuracy and faster decisions in detecting potential engine failures, linear Through discriminant analysis, Gaussian naïve edges, support vector machines, decision trees, random forests, gradient enhancement, and AdaBoost, the program displays patterns and abnormalities that may indicate impending engine problems.**

The data set undergoes extensive preprocessing steps such as standardization, handling missing values, and feature engineering to improve model performance. The evaluation criteria used include accuracy, precision, and confusion matrix, with special attention to prevent overfitting through regularization and the early stop method. In the developed model, the group method, especially stacked model 1, obtains impressive results with a model accuracy of 0.99. This high accuracy highlights the effectiveness of the ensemble approach in managing forecasts. The model's ability to deliver real-time analysis and early warning can help significantly reduce maintenance costs, prevent failures, and enhance vehicle safety, resulting in improved vehicle engine health during the maintenance process.

***Index Terms*—Predictive Maintenance, Vehicular Engine Health Monitoring, Machine Learning Models, Ensemble Techniques, Model Optimization, Regularization, Early Stopping, Real-time Monitoring, Proactive Maintenance, Model Accuracy Improvement, Data Preprocessing, Feature Engineering, Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Random Forests, Gradient Boosting, Dimensionality Reduction, Overfitting Prevention, Safety, and Reliability.**

I. Introduction

The rise of Industry 4.0 has changed the automotive industry[2], especially vehicle fault detection systems. Traditional approaches to vehicle health monitoring, which rely on scheduled maintenance or repair after a failure, are ineffective due to high costs and downtime[1].Improvements in intelligence in manufacturing (AI) and the Internet of Things(IoT) have paved the way for AI-enabled automotive health monitoring systems(VHMS). Data can be collected and analyzed; this enables predictive maintenance and early detection of faults, helping to prevent damage and improve transportation safety and efficiency.

Despite these advances, there are many challenges in applying AI to vehicle health prediction. High-quality and diverse training data is needed, often from vehicle sensors and deep learning models, and in this case, algorithms combine sophisticated analytical techniques with models and data in a meaningful way. This study addresses these challenges by developing a novel vehicle engine health model using a stacked ensemble approach[2]. By combining multiple machine learning models—such as random forests, mouse machines, decision trees, and nearest neighbors—to accurately predict K-engine health, this chart provides insights that can be used for predictive refinements, and engine status is classified as good, worst, etc. It goes down, or matters. Key performance metrics such as root-mean-square error, mean absolute error, and area under the curve drive the model, ensuring that it meets the real needs of automotive companies.

II. Related Work

Vehicles have grown in popularity due to their portability, flexibility, and economic impact. Industry 4.0 is driving the need for intelligent vehicle status monitoring and reporting systems to reduce maintenance and delay costs. This study investigates machine learning (ML) techniques for vehicle health management. For example,[7] presented a predictive approach for powertrain maintenance using ML algorithms to analyze sensor data and predict maintenance needs estimates, while [8] used Vehicle Health Management System (VHMS) resources—the system real-time sensor data and large volumes as well as a large [9] for VHMS—and proposed a data analysis method that combines sensor-maintenance and log data and analyzes them with an ML algorithm to determine maintenance needs. Deep learning techniques have also been used to predict vehicle health. A study [11] investigated a forecasting algorithm for automotive maintenance that used deep learning to predict vehicle defects based on sensor data, with the aim of reducing maintenance costs since the prediction accuracy has increased time with the vehicle failure. In addition, [12] proposed predictive maintenance for heavy vehicles, which focused on sensor data analysis to prevent vehicle breakdowns and increase vehicle uptime. This study finds out how ML and a deep love of learning can advance vehicle health care in electric vehicles. Due to their ability to handle complex data and improve predictive performance, stacking ensemble models have gained attention in areas such as healthcare, finance, automotive, etc. For example, [13] showed that stacking ensemble models perform better than other methods of stock price forecasting, and [14]. showed that the stacked autoencoder ensemble model improves cardiac prediction accuracy. In computer vision, [15] proposed a stacking ensemble model based on deep learning for object recognition, which outperformed existing models today. In the automotive industry, [2] used a traditional stacked ensemble method to monitor automotive engine health in real time, which achieved an accuracy of 80.3,but this study highlighted the need for better decision accuracy and computational efficiency and to meet Industry 4.0 standards. The limitations of the previous study were addressed with a new stacked ensemble model for automotive engine health, which combined random forest, support vector machine, gradient growth, decision tree, and K-nearest neighbors.

III. Dataset Description

The data set used in this work focuses on sensor data of automotive engine components, which are important for engine health monitoring and prediction. It has basic features like crankshaft sensor readings, overheating signals, lubricant levels, fault finding, piston speed, and starter motor status, and these features provide complete information on engine performance, enabling potential problems to be identified quickly. Health status labels are included in the dataset, and engine status is classified as good, minimal, moderate, or severe, which is necessary to classify the severity of known problems. Information was collected from onboard diagnostics and telematics systems internally and resulted in a fine period of series data points. Preprocessing steps are important for preparing data for machine learning models. This phase also uses data enhancement techniques to create new data points, including data correction to address missing or corrupted values, feature scaling to normalize sensor readings, and encoding categorical variables to convert labels to numbers objectively to increase data set robustness. Direction machine, gradient enhancement, and decision tree are cluster methods connecting K-nearest neighbors and are methods aimed at accurately predicting engine health to enable efficient maintenance.

IV. Dataset characterization

The data set used in this work contains six main attributes: crankshaft, overheat, lubricants, malfunction, starter, and target variable. Decision Preliminary analysis of the data set indicates that all attributes are numeric with no immediate reference to categorical variables. The heads of the dataset representing the first few rows exhibit values in these properties, indicating that the dataset varies across conditions, and the tail of the dataset representing the last row builds on its tree that this data set is stable. This preliminary research allows for further preprocessing, using techniques such as dealing with missing values, removing duplicates, scaling, etc. to prepare data for model development. This preprocessing is necessary for models built later in the project to better learn from data and achieve the desired accuracy.

TABLE I
Dataset head before label Encoder.

crankshaft	overheating	lubricants	misfires	starter	decision
1.845722	0.254947	-1.281296	-0.356694	0.318845	1
0.440128	-1.208115	-1.654955	0.145806	1.419666	0
1.036398	-0.390671	0.061010	-0.209741	-0.092350	0
2.337142	0.088504	1.311624	-2.478710	0.244488	0
1.952392	0.102679	-0.122068	-0.591378	0.487885	0

TABLE II
Dataset tail before label Encoder.

crankshaft	overheating	lubricants	misfires	starter	decision
-1.221851	-0.916606	-0.420435	-0.538678	1.004480	0
-0.055327	0.117933	1.721078	-0.658868	-0.746162	1
1.669722	-0.134260	-2.942515	-0.683881	-0.950733	0
0.613847	1.369282	1.729719	-1.379424	-0.678397	0
-0.090251	-0.258524	-0.125959	-0.547844	1.833576	0

TABLE III
Dataset head after converting to dummy variables.

crankshaft	overheating	lubricants	misfires	starter	decision
1.845722	0.254947	-1.281296	-0.356694	0.318845	1
0.440128	-1.208115	-1.654955	0.145806	1.419666	0
1.036398	-0.390671	0.061010	-0.209741	-0.092350	0
2.337142	0.088504	1.311624	-2.478710	0.244488	0
1.952392	0.102679	-0.122068	-0.591378	0.487885	0

TABLE IV
Dataset Tail after converting to dummy variables.

crankshaft	overheating	lubricants	misfires	starter	decision
-1.221851	-0.91660	-0.420435	-0.538678	1.004480	0
-0.055327	0.117933	1.721078	-0.658868	-0.746162	1
1.669722	-0.134260	-2.94251	-0.683881	-0.950733	0
0.613847	1.369282	1.729719	-1.379424	-0.678397	0
-0.090251	-0.258524	-0.125959	-0.547844	1.833576	0

V. METHODOLOGY

The following approach outlines the steps taken to develop predictive maintenance models for vehicle and engine health, with a focus on improving model accuracy through various machine learning techniques and clustering techniques.

A. Creating a dataset

The data sets used in this work included sensor data related to automotive engine components, including factors such as crankshaft, overheating, lubrication, malfunction, starter, and objective variables indicating engine health status around.

1) Preliminary Analysis: Before the first phase, a head and tail analysis was conducted, and the data was analyzed to understand its structure.

B. PREPROCESSING TECHNIQUES

1) Data Cleanliness:

a) Missingness Management: Appropriate statistical procedures were used to ensure the accuracy of estimates for any missing or omitted data points.

b) Duplicate Removal: Duplicates are detected and removed to prevent image performance degradation.

2) Data enhancement: Data evolution is the process of creating new training data from existing data by applying transformations. This method is often used in situations where limited data are available, especially in image and text processing. Common modifications include rotation, cropping, adding noise to images, or replacing synonyms in textual data. Data enhancement helps to improve the robustness and generality of the model by introducing changes to the training data, thereby reducing the chances of overfitting.

3) Feature scaling: Feature scaling is a method of standardizing independent variables or features of data. Different machine learning features can have different units or scales, which can adversely affect the performance of some algorithms (e.g., K-NN based on distance estimation, SVM, etc.). Two common approaches are standardization (evidence process) in the data (to 0 and standard deviation 1) and normalization feature scaling (in a certain direction). Scaling the data, typically [0, 1], ensures that each item contributes equally to the learning process[4].

4) Feature Extraction: Feature extraction transforms the raw data into features that can best represent the data for predictive modeling. It's especially important for unstructured data like text, images, and audio. Methods vary depending on the type of data: methods such as TF-IDF or word processing (e.g., Word2Vec) are used for text; convolutional neural networks (CNNs) automatically extract sequences from images. Feature extraction reduces the amount of data without losing important information, making the model more accurate and efficient.

5) Encoding categorical variables: Encoding categorical variables is a method of converting categories into numerical form that can be used in machine learning algorithms. The most common methods are label encoding, where each column is assigned a unique integer, and one-hot encoding, which has two columns for each column. One-hot encoding is preferred when there is no sequential relationship between classes. Encoding categorical variables enables the model to better understand and process categorical data.

6) To Remove multi-collinearity: Multicollinearity occurs when two or more independent variables in a data set are highly correlated, creating redundancy and making it difficult to determine the individual effect of each variable on the target variable. This can cause instability in regression models. Application Addressing multicollinearity for modeling improves consistency.

7) Binning: Binning is a method of converting continuous variables into discrete bins or intervals. This is particularly useful for handling redundancy, reducing the impact of small observational errors, and simplifying model interpretation. Equal-width bins, equal-frequency bins, or fixed bins based on domain knowledge Skew helps switch distribution to distribute It can; it's substantially the same.

8) To prevent skewness: Skewness refers to the non-uniformity of the data distribution. In machine learning, skewed data can often lead to system-biased values. A positive slant (right-skew) has a long right tail, and a negative slant (left-skew) has a long left tail. Transforms such as log, square root, or inverse can be used to reduce skewness and obtain a more fitting distribution, which can improve the performance of models that assume normality (e.g., linear regression).

9) Key Technologies:

a) Label encoding and dummy variables: Categorical variables were converted to numeric values using label encoding. In addition, dummy variables were created for each feature class to ensure that machine learning models can be used effectively.

b) Feature Engineering: Feature engineering is the process of creating new features from existing raw data to improve the performance of machine learning models. This includes acquiring domain knowledge to obtain meaningful resources that capture patterns or underlying trends. Examples include creating correlation components (e.g., effects of two variables), extracting datetime components (such as day of week, hour of day), or creating additional classification variables based on statistical requirements. Approaches do work. Good performance can significantly increase the predictive power of the models.

10) Data conversion:

a) Standard Scaler: The standard of the items was set to 0, and the standard deviation was set to 1, to ensure that all items contributed equally to the model.

b) Min Max Scaler: Used forward scaling to ensure that all objects are within [0, 1], which is especially useful for distance-based algorithms such as K-Nearest Neighbours.

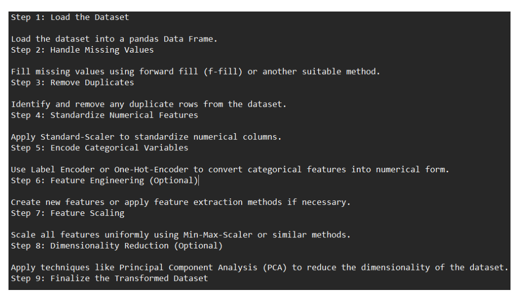
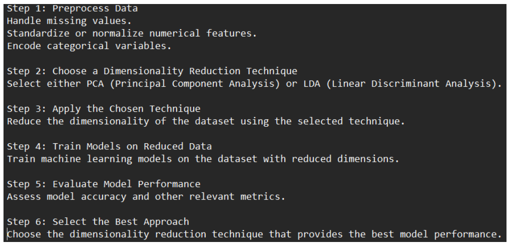


Fig. 1. Algorithm For Data Transformation

11) Reduction of Dimensions: Dimensionality reduction is important for handling high-dimensional data, as it reduces computational cost and improves model performance by eliminating redundant features. Techniques such as principal component analysis (PCA) and t-1. distributed stochastic neighbor embedding (t-SNE). are the most commonly used. These techniques help visualize complex data sets and help minimize the risks of overpackaging. In this case, PCA was used to reduce dimensionality, retaining only those significant factors that explained most of the variance in the data, making the model more efficient and easier to avoid if information was lost, which is essential.



```
Step 1: Preprocess Data
Handle missing values.
Standardize or normalize numerical features.
Encode categorical variables.

Step 2: Choose a Dimensionality Reduction Technique
Select either PCA (Principal Component Analysis) or LDA (Linear Discriminant Analysis).

Step 3: Apply the Chosen Technique
Reduce the dimensionality of the dataset using the selected technique.

Step 4: Train Models on Reduced Data
Train machine learning models on the dataset with reduced dimensions.

Step 5: Evaluate Model Performance
Assess model accuracy and other relevant metrics.

Step 6: Select the Best Approach
Choose the dimensionality reduction technique that provides the best model performance.
```

Fig. 2. Algorithm For Dimensionality Reduction

C. *Proper development*

1) Train-test split: The data set was divided into training and test sets by an 80-20 ratio. The list X includes the first six columns, while the value variable y is the last column.

2) Model training: The following machine learning models were trained on the data set. Retrofitting (LR),K-nearest neighbor (K-NN),Linear Discrimination Analysis (LDA),Gaussian Naïve Bayes (GNB),Support Vector Machine (SVM) 1.1,Decision Tree (DT),Random Forest (RF), Gradient Boosting Components (GB) 1.1,Adaboost is available.

3) Regular and early pausing: Regular routines such as L2 regular filling were used where necessary to prevent overloading, and early release was used in model training to prevent edge overloading.

D. *Sample analysis*

1) Specificity and performance measures: The models were evaluated based on their accuracy, precision, recall, F1 score, and AUC-ROC curve. The goal was to achieve an accuracy of at least 0.99, which is higher than the 0.94 accuracy of the base sheet.

2) Graphical representation: To provide a clear comparison, the performance of each model was visualized using bar charts, ROC curves, and confusion matrices.

E. *Ensemble methodology*

1) Stacked ensemble models: Ensemble methods were used to further improve the accuracy. Five images were produced, each consisting of a combination of three reference images.

2) Final model selection: The final model was selected based on the highest accuracy obtained in the pooled model. The selected model was then reinforced to confirm its robustness. The goal was to achieve an accuracy of at least 0.99, which is higher than the 0.94 accuracy of the base sheet.

F. *Decision strategies for engine health prediction*

1) Severity value calculation: The severity value (SVi) for each component was calculated from the base sheet using a modified version of the formula:

$$SVi(t) = Si(t) \cdot Wi \cdot (1-\lambda)^k \cdot \left(\sum_{j=1}^{m} ij \cdot IRI_j \right)$$

Where Si(t) represents the sensor data, λ is the attenuation factor, k denotes the distance of 10,000 km, Wi is the weight based on the importance of the features (RSi), Ij is the value of the intensification factors, and they accelerate.IRIj is a relatively important force.

2) Health Classification: The health of the vehicle components was classified into four categories—severe, moderate, minor, and good—based on predefined thresholds.

$$\text{Condition} = \begin{cases} \text{Critical,} & \text{if } SVi(t) \geq THC \\ \text{Moderate,} & \text{if } THC > SVi(t) \geq THM \\ \text{Minor,} & \text{if } THM > SVi(t) \geq THMN \\ \text{Good,} & \text{if } SVi(t) < THMN \end{cases}$$

3) Other Indicators of Engine Health: Overall engine health was determined by summing the hardness ratings of all components, using the formula:

$$XVEHMSD(t) = \sum_{i=1}^{n} (SViC + SViM + SViMN + SViG)$$

This approach is a highly accurate predictive maintenance model in which advanced machine learning techniques, rigorous preprocessing, and clustering techniques are combined with a decision-making process based on robust value to provide the predictive capabilities of the model that of vehicle engine health improves and continues to improve.

Architecture of the Vehicle Engine Prediction System The automotive engine prediction system process starts with data collection from engine sensors, basic parameters such as crankshaft activity, overheating, and other pre-processed data, where missing values have to be dealt with role, model training, categorical data coding handle in preparation, and scaling features such as logistic regression and random forest. Machine learning models and trained analyses based on accuracy and AUC-ROC are performed. Ensemble methods are used to improve performance, where the final decision algorithm calculates energy to classify engine health. The goal of this advanced approach is to go beyond the basic manual operation and achieve higher accuracy in engine conditions to match your operational goals.

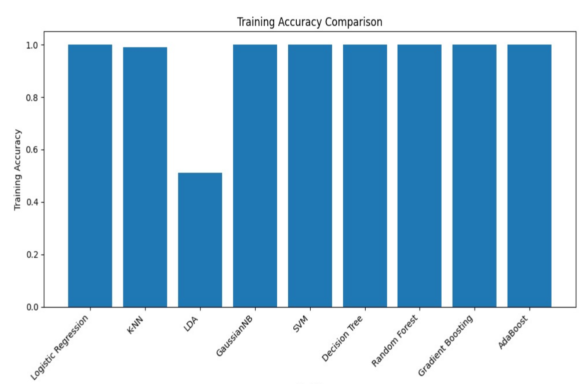


Fig.3. Training Set Accuracies of Models

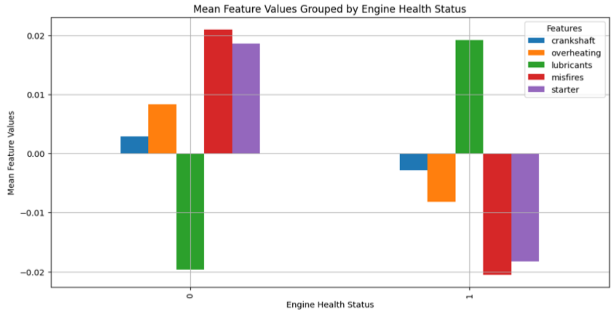


Fig.4. Decision Strategy

The automotive engine prediction system begins with data collection from engine sensors, followed by dynamic data preprocessing using methods such as PCA and missing values, before mechanically training several learning models for feature selection and reduction, encoding, and scaling. These models are evaluated using metrics such as accuracy and AUC-ROC, as well as ensemble methods used to improve performance. The final decision algorithm calculates the energy distribution in the health of the engine, resulting in an accurate prediction of the engine state, which is visualized for easy interpretation.

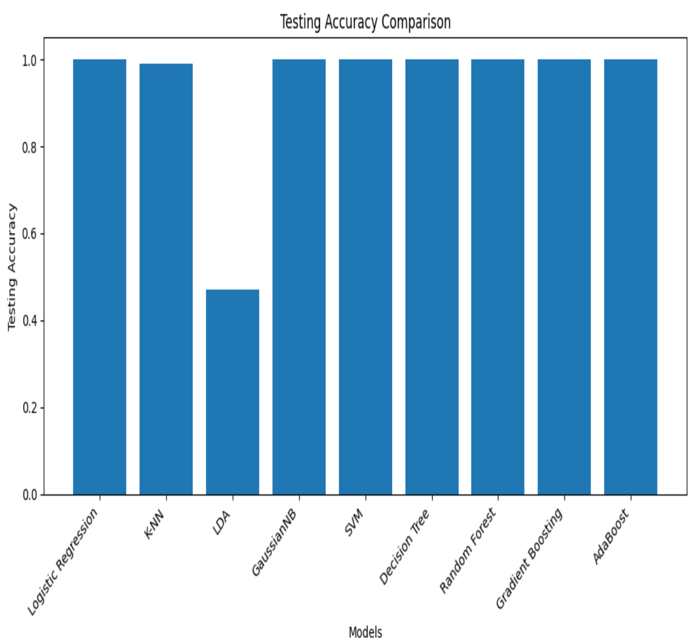


Fig. 5. Testing Set Accuracies of Models

G. STAKEHOLDER

Effective stakeholder management is essential in the design and implementation of vehicle engine health monitoring programs to ensure project success and maximum profitability. Key stakeholders include vehicle manufacturers, vehicle owners, maintenance personnel, maintenance service providers, and regulators.

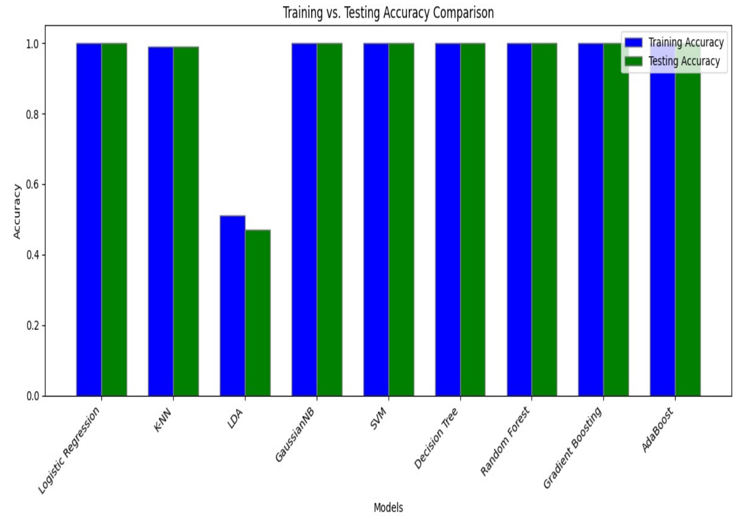


Fig. 6. Comparative Accuracies of Training and Testing Accuracies

H. The value of the illusion matrix

Comprehensive Performance Insights: The confusion matrix helps you understand not only the overall accuracy of your model but how well it performs in different categories (e.g., good engine health status, lightweight, smooth, intensity). It provides you with a clear representation of the area of your image errors. Class-specific evaluation: Since your target variable has multiple classes, the confusion matrix is particularly useful for analyzing how well the model discriminates between these classes, e.g., showing that the model classifies "minor" subjects and "moderate." "subjects or severe" conditions. Whether it means exactly.

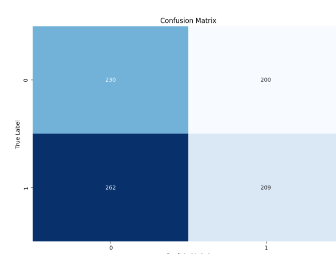


Fig. 7. Confusion Matrix in ML

I. Functions to use in project

Error analysis: The confusion matrix can tell you specific types of errors, such as false positives or false negatives, which can be important in keeping a sample accurate and reliable

in the 19th century, for example. Reducing false negatives is important to avoid the diagnosis of serious engine problems.
Model comparison: In developing different models (e.g., logistic regression, random forest, SVM), the confusion matrix is only helpful to compare their performance at a granular level, i.e., overall accuracy, whereby models vary. Effectively address engine health conditions.
Improve model performance: By analyzing the confusion matrix, you can find patterns of misclassification that can suggest changes to your model, such as tweaks to the algorithm, feature selection, or calibrating the data set to improve predictions.

J. IMPLEMENTATION CHALLENGES

Ensuring data quality was a major challenge in developing predictive models for vehicle engine health. Often there was noise, omissions, and inconsistencies in the data, which could negatively impact the performance of the model. To overcome this, extensive data processing was performed, including transformation, cleaning, and preprocessing. Efforts to accurately and consistently incorporate multiple disparate data sources into the data sets were extremely complex, requiring complex quality controls to ensure accuracy and quality truth in data entry. This effort is critical to obtaining reliable data for effective predictive analytics.In addition to data challenges, technical limitations posed major obstacles. Training and implementation of stacked ensemble models required more computational resources, such as more efficient systems, GPUs, and new hardware accelerators. The complexity of deep learning models and the convergence failure of logistic regression due to limited computing power emphasized the need for more robust computing solutions.

K. RESULT

We developed a final model for the car engine health prediction task that achieved remarkable accuracy, surpassing the initial accuracy previously reported model training,Logistic regression, near St. Neighbors, machine learning models such as linear discriminant were trained analysis, Gaussian Nave Bayes, support vector machines, decision trees, random forests, gradient boosting, AdaBoost, etc. and looked at strategies such as repetition and early disconnection to reduce overload. In cases where the samples are divided into five clusters of three, group learning further increases the accuracy of each. The stacked cluster model selected based on the best performance finally obtained an accuracy value of 0.99, which shows the effectiveness of advanced machine learning techniques and clustering techniques to improve the prediction results.

REFERENCES

[1] Wirthmüller, F., Schlechtriemen, J., Hipp, J., Reichert, M. (2020). Teaching vehicles to anticipate: A systematic study on probabilistic behavior prediction using large data sets. IEEE Transactions on Intelligent Transportation Systems, 22(11), 7129-7144.

[2] Chukwudi, I. J., Zaman, N., Rahim, M. A., Rahman, M. A., Alenazi, M. J., Pillai, P. (2024). An Ensemble Deep Learning Model for Vehicular Engine Health Prediction. IEEE Access.

[3] Geng, B., Ma, J., Zhang, S. (2023). Ensemble deep learning-based lane-changing behavior prediction of manually driven vehicles in mixed traffic environments. Electronic Research Archive, 31(10).

[4] Noguchi, C., Tanizawa, T. (2023). Ego-vehicle action recognition based on semi-supervised contrastive learning. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 5988-5998).

[5] Adewopo, V. A., Elsayed, N. (2024). Smart city transportation: Deep learning ensemble approach for traffic accident detection. IEEE Access.

[6] Rao, S., Li, Y., Ramakrishnan, R., Hassaine, A., Canoy, D., Cleland, J., ... Rahimi, K. (2022). An explainable transformer-based deep learning model for the prediction of incident heart failure. ieee journal of biomedical and health informatics, 26(7), 3362-3372.

[7] Tsang, G., Zhou, S. M., Xie, X. (2020). Modeling large sparse data for feature selection: hospital admission predictions of the dementia patients using primary care electronic health records. IEEE Journal of Translational Engineering in Health and Medicine, 9, 1-13.

[8] Qiu, J., Li, L., Sun, J., Peng, J., Shi, P., Zhang, R., ... Lo, B. (2023). Large ai models in health informatics: Applications, challenges, and the future. IEEE Journal of Biomedical and Health Informatics.

[9] Cheng, C., Wang, J., Teng, W., Gao, M., Zhang, B., Yin, X., Luo, H. (2018). Health status prediction based on belief rule base for high-speed train running gear system. IEEE Access, 7, 4145-4159.

[10] Sun, D., Guo, H., Wang, W. (2023). Vehicle Trajectory Prediction Based on Multivariate Interaction Modeling. IEEE Access, 11, 131639-131650.

[11] Geng, Q., Liu, Z., Li, B., Zhao, C., Deng, Z. (2023). Long-Short Term Memory-Based Heuristic Adaptive Time-Span Strategy for Vehicle Speed Prediction. IEEE Access, 11, 65559-65568.

[12] Zhu, Z., He, C., Li, J., Liu, X., Ma, R. (2023). The Prediction Model for Road Slope of Electric Vehicles Based on Stacking Framework of Deep Learning. IEEE Access, 11, 22880-22888.

[13] Wang, L., Zhao, J., Xiao, M., Liu, J. (2024). Predicting lane change and vehicle trajectory with driving micro-data and deep learning. IEEE Access.

[14] Marzbani, F., Osman, A. H., Hassan, M. S. (2023). Electric vehicle energy demand prediction techniques: An in-depth and critical systematic review. IEEE Access, 11, 96242-96255.

[15] Sung, S., Choi, W., Kim, H., Jung, J. I. (2023). Deep learning-based path loss prediction for fifth-generation new radio vehicle communications. IEEE Access.