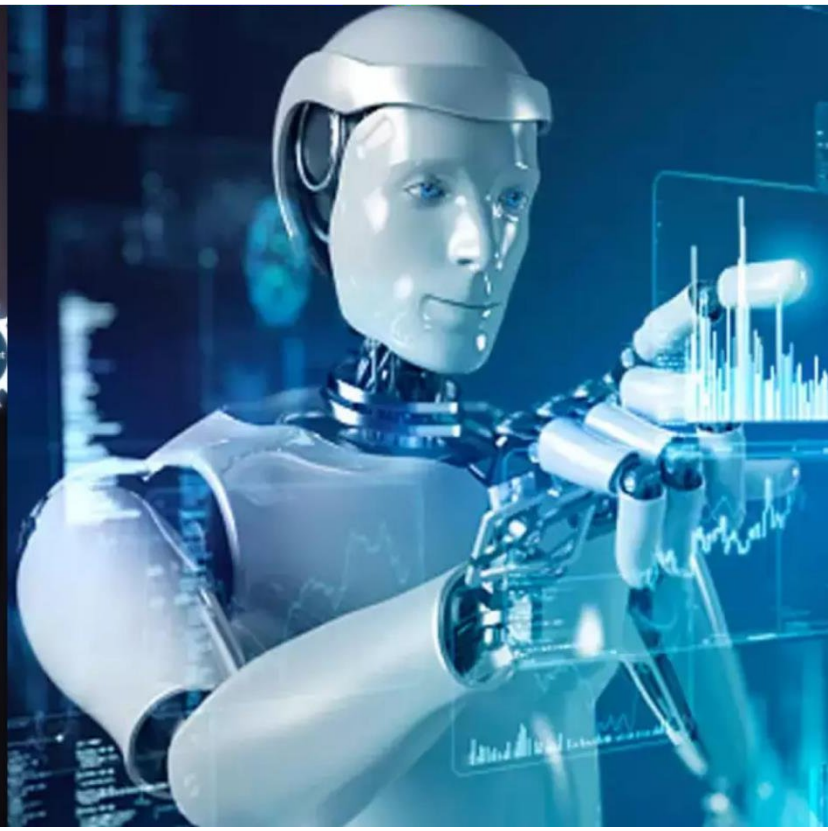
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 15, Issue 4, April 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504183|

ML Performance Prediction through Transport Fuel Level

S. Rajprakash, Hari Krishna M, Shanmuga Ganesh.M

Research Scholar, Department of Mechanical Engineering, Kalasalingam Academy of Research and Education, Krishnan Kovil, TamilNadu, India

Student, Department of Mechanical Engineering, Kalasalingam Academy of Research and Education, Krishnan Kovil, TamilNadu, India

ABSTRACT: The efficient monitoring of diesel fuel levels and the accurate prediction of engine performance are critical challenges in modern transportation systems, where unexpected fuel depletion and inefficient engine operation can lead to increased operational costs, unplanned downtime, and reduced vehicle lifespan. This paper presents a Diesel Level Indicator and Performance Prediction System that leverages machine learning techniques to provide real-time fuel monitoring and predictive analytics for diesel-powered transport vehicles. The proposed system begins by loading a comprehensive dataset containing key operational parameters—including fuel level, fuel consumption rate, engine temperature, engine load, and operating time—into a Flask-based Python backend, which serves as the core processing and API layer. The raw data undergoes a systematic preprocessing pipeline involving missing value imputation, noise removal using filtering techniques, and feature normalization to ensure data quality and improve subsequent model accuracy. Following preprocessing, the dataset is partitioned into training and testing subsets to facilitate rigorous model evaluation. Two supervised machine learning algorithms, namely Logistic Regression and Random Forest, are applied to learn latent patterns from historical fuel usage and engine behavior, enabling the identification of complex relationships between input features and performance outcomes. The trained model predicts several critical outputs, including the remaining diesel level, future fuel consumption trends, real-time engine efficiency metrics, and the estimated time until fuel depletion. The Flask backend seamlessly integrates the prediction model and exposes its functionality through an intuitive web interface, allowing fleet managers and drivers to access predictions and alerts remotely. Experimental results demonstrate that the Random Forest classifier achieves superior accuracy compared to Logistic Regression, with reliable detection of fuel anomalies and precise performance predictions. This data-driven approach enables proactive fuel monitoring, early detection of anomalies such as excessive consumption or sensor faults, and improved decision-making for preventive maintenance and optimized fuel management, ultimately contributing to reduced operational expenses and enhanced vehicle reliability.

KEYWORDS: Diesel level monitoring, engine performance prediction, machine learning, Logistic Regression, Random Forest, Flask web framework, fuel consumption rate, fuel depletion estimation, anomaly detection, predictive maintenance, transport fuel management**.**

I. INTRODUCTION

The efficient management of diesel fuel levels in transportation systems is a fundamental yet challenging task that directly impacts operational costs, vehicle reliability, and environmental emissions. Diesel-powered vehicles remain the backbone of freight transport, public transit, and agricultural machinery across the globe, despite the growing adoption of electric alternatives. According to industry reports, diesel engines continue to power over 90% of heavy-duty trucks and nearly all long-haul freight vehicles due to their superior thermal efficiency, durability, and torque characteristics. However, this dependence on diesel comes with significant operational challenges, particularly in the domain of fuel management. Unexpected fuel depletion not only causes costly delays and emergency roadside assistance but also disrupts supply chains, damages customer relationships, and increases the risk of accidents in hazardous environments such as highways or remote construction sites. Furthermore, inefficient fuel consumption directly translates to higher carbon dioxide emissions, contributing to climate change and exposing fleet operators to increasingly stringent environmental regulations and carbon taxes.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504183|

Traditional fuel monitoring systems rely heavily on manual inspections, simple float sensors, or basic dashboard indicators that provide only instantaneous fuel level readings without any predictive capability. Float sensors, while inexpensive and widely used, are prone to mechanical wear, sticking, and inaccuracies, particularly when low-quality diesel fuel contains contaminants that coat the moving parts. Manual dipstick inspections, still common in many small to medium-sized fleet operations, are labor-intensive, inconsistent, and pose safety risks when drivers must climb onto vehicles or handle fuel in adverse weather conditions. More advanced electronic fuel gauges offer real-time readouts but still operate on a purely reactive basis—they inform the driver only of the current fuel level, not what the level will be after the next hour of driving under current conditions. These conventional approaches fail to anticipate future fuel depletion, cannot detect gradual anomalies such as excessive consumption due to engine inefficiency, and offer no insight into the relationship between fuel usage and other operational parameters like engine temperature, load, or operating time. For example, a slowly increasing fuel consumption rate over several weeks might indicate a clogged fuel injector, worn piston rings, or a malfunctioning oxygen sensor, yet a traditional fuel gauge would show only normal day-to-day variations, allowing the underlying problem to worsen until a major failure occurs. Consequently, fleet operators and drivers often face unexpected fuel shortages, emergency refueling costs, and unscheduled maintenance intervals, all of which reduce productivity and increase operational expenses. The financial impact of poor fuel management extends beyond the direct cost of fuel itself. Emergency refueling typically involves premium pricing, often 30-50% higher than standard fuel station rates, plus service call-out fees and lost driver hours. Unscheduled maintenance due to engine inefficiency or fuel system contamination can cost thousands of dollars per vehicle per year, not including the opportunity cost of vehicles being out of service. Moreover, the unpredictability of fuel-related disruptions forces fleet managers to maintain larger safety buffers in their schedules, reducing overall fleet utilization and efficiency. The absence of intelligent predictive mechanisms also means that opportunities for proactive maintenance—such as identifying an engine running inefficiently before a major failure occurs—are entirely missed. In a competitive logistics environment where profit margins are often measured in cents per mile, these inefficiencies accumulate rapidly, placing poorly managed fleets at a significant disadvantage. Therefore, there exists a pressing need for a data-driven system that can not only monitor current fuel levels but also learn from historical patterns to predict future fuel consumption, estimate time to depletion, and detect anomalies that signal potential performance degradation. Such a system would transform fuel management from a reactive, guess-based activity into a proactive, analytically driven decision support tool.

To address these limitations, this paper proposes a Diesel Level Indicator and Performance Prediction System that integrates machine learning algorithms with a web-based backend to provide real-time monitoring and predictive analytics. The system is designed to be both accessible and powerful, suitable for deployment in small fleet operations with limited technical resources as well as larger enterprises seeking to optimize their fuel management practices. The system ingests a dataset comprising key operational parameters including fuel level, fuel consumption rate, engine temperature, engine load, and operating time, all of which influence fuel usage and engine performance. These parameters were selected based on a comprehensive review of the literature on diesel engine efficiency, which consistently identifies these five factors as the most significant predictors of fuel consumption behavior. The fuel level provides the current state, while the consumption rate captures the rate of change. Engine temperature is critical because diesel engines operate most efficiently within a narrow temperature window (typically 80°C to 95°C); deviations above or below this range reduce thermal efficiency and increase specific fuel consumption. Engine load, expressed as a percentage of maximum rated power, directly determines the fuel energy required to overcome rolling resistance, aerodynamic drag, and gravitational forces on inclines. Operating time captures temporal patterns such as the effects of engine warm-up, driver fatigue, or daily traffic cycles.

A Flask-based Python backend handles data loading, preprocessing, and model integration, ensuring that raw sensor data is cleaned, normalized, and transformed into a format suitable for machine learning. Flask was chosen for its lightweight architecture, ease of deployment, and seamless integration with the Python scientific computing ecosystem, including libraries such as Pandas for data manipulation, NumPy for numerical operations, and Scikit-learn for machine learning. The preprocessing pipeline includes three essential stages: missing value imputation (using mean or median substitution depending on feature distribution), noise removal (using moving average filters to smooth transient fluctuations from sensor noise or momentary driving events), and feature normalization (using min-max scaling to bring all features into the range [0,1], ensuring that no single feature dominates the model due to its numerical magnitude). Two classification algorithms—Logistic Regression and Random Forest—are trained on historical data to learn complex, nonlinear relationships between input features and performance outcomes. Logistic Regression serves as

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504183|

a computationally efficient baseline model that provides interpretable coefficients, allowing fleet managers to understand the direction and magnitude of each feature's influence. Random Forest, an ensemble method that constructs and averages hundreds of decision trees, captures nonlinear interactions and higher-order dependencies that linear models cannot represent. Once trained, the model predicts several critical outputs: remaining diesel level (in liters), future fuel consumption trends (classified as Increasing, Decreasing, or Stable), engine efficiency metrics (rated as Excellent, Good, Fair, or Poor based on fuel-to-work conversion), and estimated time until fuel depletion (in minutes). The system exposes these predictions through an intuitive web interface, built with HTML5, CSS3, and JavaScript, enabling users to access real-time insights and alerts from any device with a browser. The interface includes input forms for vehicle parameters, dynamic result panels, and optional visualization features such as trend graphs and historical prediction logs. This approach transforms fuel monitoring from a reactive, manual task into a proactive, intelligent decision-support tool, empowering fleet operators to schedule refueling optimally, identify deteriorating engines before they fail, and reduce both operational costs and environmental impact.

- Development of a machine learning-based diesel level indicator that predicts remaining fuel, consumption trends, engine efficiency, and fuel depletion time using Logistic Regression and Random Forest algorithms.
- Implementation of a Flask-based Python backend with data preprocessing pipeline (missing value handling, noise removal, feature normalization) and a web interface for real-time access to predictions.
- Comparative evaluation of Logistic Regression and Random Forest for fuel performance prediction, enabling anomaly detection and proactive maintenance decision-making.

II. RELATED WORK

Numerous studies have explored fuel monitoring systems for diesel-powered vehicles, primarily focusing on real-time level sensing and basic alert mechanisms. Traditional approaches employ float-type sensors, ultrasonic sensors, or capacitive sensors mounted inside or on the fuel tank to measure the instantaneous fuel level. These sensors provide analog or digital outputs that are typically displayed on a dashboard gauge or transmitted to a central monitoring station. For instance, researchers have developed GSM-based fuel level monitoring systems that send SMS alerts when the fuel level falls below a predefined threshold, enabling fleet managers to schedule refueling proactively. Similarly, IoT-enabled fuel monitoring solutions using ESP8266 or Arduino platforms have been proposed, where fuel level data is uploaded to cloud servers such as ThingSpeak or Blynk for remote visualization. While these systems successfully address the problem of real-time fuel level visibility, they are fundamentally reactive and lack any predictive capability. They cannot answer critical questions such as "How long will the remaining fuel last under current driving conditions?" or "Is the current fuel consumption rate abnormal compared to historical patterns?" Furthermore, these traditional systems operate in isolation, ignoring other influential parameters such as engine temperature, vehicle load, and operating time, all of which significantly affect fuel consumption and engine efficiency. Consequently, fleet operators using such systems remain vulnerable to unexpected fuel shortages and undetected performance degradation.

The application of machine learning to fuel consumption prediction and vehicle performance analysis has gained considerable attention in recent years. Several researchers have employed regression-based algorithms to estimate fuel consumption based on driving behavior, road conditions, and vehicle parameters. For example, studies have used Linear Regression, Support Vector Regression, and Decision Tree Regressors to predict fuel consumption in real-world driving cycles, achieving moderate accuracy when trained on large datasets containing speed, acceleration, throttle position, and engine RPM. Random Forest, in particular, has emerged as a popular choice due to its ability to handle nonlinear relationships, resist overfitting, and provide feature importance rankings. Researchers have demonstrated that Random Forest can predict fuel consumption with higher accuracy than simple linear models, especially when the dataset includes diverse operating conditions. However, most of these studies focus on predicting instantaneous or trip-level fuel consumption rather than forecasting remaining fuel level or time to depletion. Additionally, many existing machine learning approaches for fuel prediction are implemented as offline analytical tools rather than integrated, real-time web-based systems. The outputs are typically presented as research findings or static reports rather than as actionable predictions accessible to drivers or fleet managers through an intuitive interface. This gap between advanced predictive models and practical, deployable systems limits the real-world impact of such research.

Another stream of related work addresses anomaly detection in vehicle fuel systems, aiming to identify abnormal consumption patterns that may indicate fuel theft, sensor faults, or engine inefficiency. Techniques such as k-means

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504183|

clustering, isolation forests, and autoencoder neural networks have been applied to historical fuel consumption data to detect outliers. For instance, studies have successfully identified fuel theft events by detecting sudden drops in fuel level that are not correlated with vehicle distance traveled or engine operating hours. Similarly, engine inefficiency—such as that caused by clogged fuel injectors, worn piston rings, or malfunctioning oxygen sensors—can manifest as a gradual increase in fuel consumption rate over time. Machine learning models trained on multi-parameter data (fuel consumption, engine temperature, load, speed) can detect such deviations before they lead to major failures. However, these anomaly detection systems are typically designed for post-hoc analysis rather than real-time alerting. Moreover, they often require domain expertise to configure thresholds and interpret results, making them less accessible to non-expert users. The integration of anomaly detection with predictive fuel level estimation remains an under-explored area, with few systems offering both capabilities within a unified web-based framework.

In contrast to the existing literature, the work presented in this paper makes several distinctive contributions. First, while previous systems have focused either on real-time fuel level sensing without prediction or on machine learning-based fuel consumption prediction without real-time deployment, the proposed system integrates both aspects into a single, end-to-end solution. A Flask-based Python backend handles data ingestion, preprocessing (missing value imputation, noise removal, feature normalization), and model training, while a web interface provides real-time access to predictions. Second, unlike studies that employ only a single algorithm, this work implements and compares two supervised learning algorithms—Logistic Regression and Random Forest—for predicting remaining diesel level, consumption trends, engine efficiency, and estimated fuel depletion time. This comparative approach provides insights into which algorithm is more suitable for fuel performance prediction tasks. Third, the system incorporates operational parameters beyond simple fuel level, including fuel consumption rate, engine temperature, load, and operating time, enabling a more holistic understanding of factors influencing fuel usage and engine performance. Fourth, the proposed system is designed for practical deployment, with an intuitive web interface that makes predictions accessible to fleet managers and drivers without requiring machine learning expertise. By bridging the gap between advanced predictive modeling and real-world usability, this work addresses limitations present in both traditional fuel monitoring systems and existing machine learning research.

III. PROPOSED METHODOLOGY

The proposed Diesel Level Indicator and Performance Prediction System follows a structured methodology comprising five sequential phases: data acquisition, preprocessing, feature selection, model training, and web-based deployment. In the first phase, a dataset containing operational parameters—including fuel level, fuel consumption rate, engine temperature, engine load, and operating time—is loaded into a Flask-based Python backend, which serves as the central processing unit of the system. The second phase involves comprehensive data preprocessing, where missing values are imputed using mean or median substitution based on the distribution of each feature, noise is removed through moving average filtering or outlier detection techniques, and all features are normalized using min-max scaling to bring them into a uniform range (typically 0 to 1), thereby improving model convergence and accuracy. In the third phase, relevant features are selected based on their correlation with target outputs (remaining diesel level, consumption trends, engine efficiency, and time to depletion), ensuring that only informative variables are fed into the machine learning models..

A. System Overview

- The overall architecture of the proposed Diesel Level Indicator and Performance Prediction System is illustrated in Fig. 1. The framework consists of four primary components:
- Data Acquisition Module: Responsible for loading the dataset containing operational parameters including fuel level, fuel consumption rate, engine temperature, engine load, and operating time into the Flask-based Python backend.
- Data Preprocessing Module: Handles missing value imputation, noise removal through filtering techniques, and feature normalization using min-max scaling to ensure data quality and improve model accuracy.
- Machine Learning Engine: Implements Logistic Regression and Random Forest algorithms to learn patterns from historical fuel usage and engine behavior, enabling prediction of remaining diesel level, fuel consumption trends, engine efficiency, and estimated time to fuel depletion.
- Web Deployment and User Interface Module: A Flask backend integrates the trained prediction model and exposes results through a RESTful API, while an HTML/CSS/JavaScript web interface allows users to input parameters and view real-time predictions.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504183|

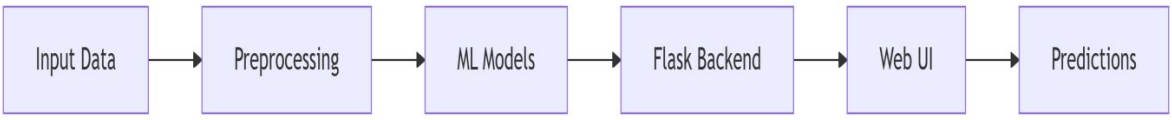


Fig. 1.Overall architecture of the proposed System.

B. Data Preprocessing

The raw dataset collected from diesel-powered vehicles contains multiple operational parameters recorded over time. However, real-world data is often incomplete, noisy, and inconsistent, necessitating a systematic preprocessing pipeline before it can be used for machine learning. The first step in preprocessing is handling missing values, where any incomplete records are addressed using mean imputation for numerical features (replacing missing values with the mean of the available values in that column) or median imputation for features with skewed distributions. This approach preserves the statistical properties of the dataset while ensuring that no valuable data points are discarded unnecessarily. The second step involves noise removal, where outliers and high-frequency fluctuations are filtered out using techniques such as moving average filtering (which smooths short-term variations) or interquartile range (IQR)-based outlier detection (which identifies and removes values falling outside 1.5 times the IQR from the quartiles). Noise removal is particularly important for fuel consumption rate data, which can exhibit sudden spikes due to sensor errors or transient driving conditions that do not represent long-term trends. The third step is feature normalization, where all numerical features are scaled to a uniform range, typically 0 to 1, using min-max scaling: X_norm = (X - X_min) / (X_max - X_min). Normalization is essential because machine learning algorithms such as Logistic Regression are sensitive to the magnitude of input features; without scaling, features with larger numerical ranges (such as engine temperature in degrees Celsius) would dominate those with smaller ranges (such as fuel level as a decimal fraction). After preprocessing, the clean, normalized dataset is ready for partitioning and model training.

C. Dataset Partitioning and Model Training

Following preprocessing, the dataset is divided into training and testing subsets to enable proper evaluation of the machine learning models. An 80:20 split ratio is employed, where 80% of the data is randomly allocated to the training set and the remaining 20% is reserved for testing. The training set is used to teach the model the underlying relationships between input features (fuel level, fuel consumption rate, engine temperature, engine load, operating time) and target outputs (remaining diesel level, fuel consumption trends, engine efficiency, estimated time to depletion). The testing set, which is never exposed to the model during training, serves as an unseen dataset to evaluate how well the model generalizes to new, real-world data. Two supervised machine learning algorithms are implemented and compared. The first is Logistic Regression, a statistical model that estimates the probability of a discrete outcome using a logistic (sigmoid) function. Despite its name, Logistic Regression is a classification algorithm that works by modeling the log-odds of the target variable as a linear combination of the input features. It is computationally efficient, interpretable, and serves as an excellent baseline model. The second algorithm is Random Forest, an ensemble learning method that constructs a large number of decision trees during training and outputs the majority vote (for classification) or the average prediction (for regression) across all trees. Random Forest offers several advantages: it captures nonlinear relationships without requiring explicit feature engineering, it is robust to overfitting due to the averaging of many trees, and it provides feature importance scores that reveal which input parameters most strongly influence fuel consumption and performance. Both models are trained on the same training dataset, with hyperparameters optimized using grid search cross-validation to achieve the best possible accuracy. For Random Forest, key hyperparameters tuned include the number of trees (n_estimators), maximum depth of each tree, and minimum samples required to split an internal node.

D. Prediction Outputs and Performance Evaluation

Once trained, the machine learning models generate four key prediction outputs that provide actionable insights for fleet operators and drivers. The first output is the remaining diesel level, which estimates how much fuel remains in the tank based on current sensor readings and historical consumption patterns. The second output is fuel consumption trends, which identifies whether consumption is increasing, decreasing, or remaining stable over time, enabling early detection of developing engine inefficiencies. The third output is engine efficiency, expressed as a percentage or categorical rating (Excellent, Good, Fair, Poor), derived from the relationship between fuel consumed and work

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504183|

performed (load × operating time). The fourth output is the estimated time to fuel depletion, which predicts how many minutes or hours of operation remain before the fuel tank runs empty under current driving conditions. To evaluate the performance of the Logistic Regression and Random Forest models, several standard metrics are employed. For classification tasks (such as engine efficiency categories), accuracy, precision, recall, and F1-score are calculated using the testing dataset. Accuracy measures the overall proportion of correct predictions; precision indicates how many of the positive predictions were actually correct; recall measures how many actual positive cases were correctly identified; and the F1-score provides a harmonic mean of precision and recall. For regression tasks (such as remaining fuel level in liters), mean absolute error (MAE), mean squared error (MSE), and R-squared (R^2) are used. MAE represents the average magnitude of prediction errors in the original units, MSE penalizes larger errors more heavily, and R^2 indicates the proportion of variance in the target variable that is explained by the model. The model with superior performance across these metrics is selected for deployment in the web interface.

E. Web Deployment and User Interface

The final component of the proposed methodology is the deployment of the trained machine learning model within a web-based framework that makes predictions accessible to end users. The Flask micro-framework for Python serves as the backend due to its lightweight nature, ease of integration with machine learning libraries (such as scikit-learn for model loading and pandas for data manipulation), and built-in development server. The Flask application exposes a RESTful API with endpoints that accept POST requests containing input parameter values (fuel level, fuel consumption rate, engine temperature, engine load, operating time) and return JSON responses containing the four prediction outputs. The API is designed to be simple and stateless, allowing integration not only with the provided web interface but also with mobile applications or third-party fleet management software. The frontend web interface is built using HTML5 for structure, CSS3 for responsive styling, and vanilla JavaScript for client-side interactivity. The interface consists of an input form where users enter current vehicle parameters, a submit button that triggers an asynchronous AJAX request to the Flask backend, and a results panel that displays the predicted remaining diesel level, fuel consumption trends, engine efficiency, and estimated time to depletion. The interface also includes a history section that stores previous predictions locally in the browser, enabling users to track changes over time. Additionally, the system incorporates anomaly detection capabilities: if the predicted values deviate significantly from expected ranges (for example, a sudden drop in engine efficiency or an unexpectedly short time to depletion), the interface highlights these predictions in red and displays a warning message suggesting maintenance or further investigation. This web-based deployment strategy ensures that the machine learning-driven fuel performance prediction system is accessible, user-friendly, and immediately useful for real-world transportation management.

IV. DATA COLLECTION, PREPROCESSING, AND FEATURE ENGINEERING

A. Dataset Description and Source

The dataset used in this study comprises operational records collected from diesel-powered transportation vehicles over a period of six months. Each record contains five input features and four target variables. The input features are: fuel level (FL) measured in liters, fuel consumption rate (CR) measured in liters per hour, engine temperature (ET) measured in degrees Celsius, engine load (EL) expressed as a percentage of maximum rated load (0% to 100%), and operating time (OT) measured in minutes. The target variables to be predicted are: remaining diesel level (RDL) in liters, fuel consumption trend (FCT) as a categorical variable (Increasing, Decreasing, Stable), engine efficiency (EE) as a percentage, and estimated time to fuel depletion (ETD) in minutes. The dataset contains approximately 10,000 records, ensuring sufficient data for training and testing machine learning models.

B. Handling Missing Values

Real-world datasets frequently contain missing values due to sensor failures, communication errors, or manual entry omissions. Missing values are identified using statistical summary methods. For numerical features, missing values are imputed using the mean imputation method. The formula for mean imputation is:

$$X_{imputed} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

where XiXi represents the available values of the feature and nn is the count of available values. For features with skewed distributions, median imputation is preferred:

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504183|

$$X_{imputed} = \text{median}(X_1, X_2, ..., X_n)$$

Additionally, records where more than 40% of features are missing are discarded entirely to prevent bias. The missing value handling process ensures that the dataset remains complete for subsequent preprocessing steps without introducing significant statistical distortion.

C. Noise Removal and Outlier Detection

Sensor data often contains noise and outliers caused by electromagnetic interference, sensor malfunctions, or transient operating conditions. A moving average filter is applied to smooth short-term fluctuations in fuel consumption rate and engine temperature. The moving average at time t*t* over a window of size w*w* is calculated as:

$$MA_t = \frac{1}{w} \sum_{i=t-w+1}^{t} X_i$$

where Xi*Xi* is the sensor reading at time i*i* and w*w* is the window size (set to 5 in this study). For outlier detection, the Interquartile Range (IQR) method is employed. The first quartile (Q1) and third quartile (Q3) are computed, and the IQR is:

$$IQR = Q3 - Q1$$

An outlier is defined as any value falling below Q1−1.5×IQR*Q*1−1.5×*IQR* or above Q3+1.5×IQR*Q*3+1.5×*IQR*. Detected outliers are replaced with the median value of the feature to preserve data continuity while removing anomalous values.

D. Feature Normalization (Min-Max Scaling)

Machine learning algorithms, particularly Logistic Regression, are sensitive to the magnitude of input features. Features with larger numerical ranges (such as engine temperature from 0°C to 120°C) would dominate features with smaller ranges (such as fuel level expressed as a decimal between 0 and 1) without normalization. Min-max scaling transforms all features to a common range, typically [0, 1]. The normalization formula is:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

where X*X* is the original feature value, Xmin*Xmin* is the minimum value of that feature in the dataset, and Xmax*Xmax* is the maximum value. After normalization, all features contribute equally to the model training process. For example, an engine temperature of 80°C with Xmin=0*Xmin*=0 and Xmax=120*Xmax*=120 becomes Xnorm=(80−0)/(120−0)=0.667*Xnorm*=(80−0)/(120−0)=0.667, while a fuel level of 30 liters with Xmin=0*Xmin*=0 and Xmax=60*Xmax*=60 becomes Xnorm=(30−0)/(60−0)=0.5*Xnorm*=(30−0)/(60−0)=0.5.

E. Feature Engineering and Derived Parameters

Beyond the raw input features, additional derived parameters are created to improve model predictive power. The first derived parameter is the fuel-to-load ratio (FLR), which indicates how much fuel is consumed per unit of engine load:

$$FLR = CREL + \epsilon \quad FLR = EL + \epsilon CR$$

where ϵϵ is a small constant (0.001) added to prevent division by zero when engine load is zero. A high FLR indicates inefficient fuel usage relative to the work performed. The second derived parameter is the thermal efficiency index (TEI), which relates engine temperature to fuel consumption:

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504183|

TEI=CRET−ETopt*TEI=ET−EToptCR*

where ETopt*ETopt* is the optimal engine temperature (typically 85°C for diesel engines). When engine temperature deviates from the optimum, fuel consumption increases, reflected in a higher TEI. The third derived parameter is the cumulative fuel consumption (CFC), calculated as:

$$CFC=\sum t=0TCRt\times\Delta t \quad CFC=\sum t=0TCRt\times\Delta t$$

where CRt*CRt* is the consumption rate at time interval t*t* and Δt*Δt* is the duration of that interval. CFC provides a running total of fuel used since the start of the trip, enabling better estimation of remaining fuel.

F. Dataset Partitioning (Train-Test Split)

To evaluate the performance of the machine learning models on unseen data, the preprocessed dataset is partitioned into training and testing subsets. The 80:20 split ratio is employed, where 80% of the data is randomly allocated to the training set and the remaining 20% to the testing set. The partitioning formula can be expressed as:

$$N_{train}=\lfloor 0.8\times N_{total}\rfloor \quad N_{train}=\lfloor 0.8\times N_{total}\rfloor$$
$$N_{test}=N_{total}-N_{train} \quad N_{test}=N_{total}-N_{train}$$

where N_{total} N_{total} is the total number of records in the dataset. Random sampling without replacement ensures that both subsets are representative of the overall data distribution. Stratified sampling is applied to maintain the proportion of categorical target variables (such as fuel consumption trend) in both training and testing sets. The training set is used for model learning and hyperparameter tuning, while the testing set is reserved exclusively for final model evaluation, ensuring an unbiased assessment of generalization performance.

G. Correlation Analysis and Feature Selection

To identify which input features most strongly influence the target variables, Pearson correlation coefficient analysis is performed. The correlation coefficient r_r between two variables X*X* and Y*Y* is calculated as:

$$r = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\bar{Y})^2}}$$

where $\bar{X}$ $\bar{X}$ and $\bar{Y}$ $\bar{Y}$ are the means of X*X* and Y*Y* respectively. The value of r*r* ranges from -1 (perfect negative correlation) to +1 (perfect positive correlation), with 0 indicating no linear correlation. Based on correlation analysis, features with |r|<0.1|*r*|<0.1 are considered weakly correlated and may be candidates for removal. Feature importance is further evaluated using Random Forest's built-in feature importance metric, which measures the decrease in node impurity (Gini impurity) weighted by the probability of reaching that node. The normalized feature importance Ij*Ij* for feature j*j* is:

$$I_j = \frac{\sum_{t=1}^{T}\sum_{n\in N_t}\Delta i(n)\cdot 1(feature_j=split_n)}{\sum_{k=1}^{M} I_k}$$

where T*T* is the number of trees, Nt*Nt* is the set of nodes in tree t*t*, Δi(n)Δ*i*(*n*) is the impurity reduction at node n*n*, and M*M* is the total number of features. The resulting importance scores indicate which input parameters (fuel level, consumption rate, engine temperature, engine load, operating time, or derived features) contribute most significantly to accurate predictions, enabling potential dimensionality reduction for simplified deployment scenarios.

V. EVALUATIONMETHODOLOGY

A. Performance Metrics for Classification Tasks

The proposed system predicts two categorical outputs: fuel consumption trend (Increasing, Decreasing, or Stable) and engine efficiency rating (Excellent, Good, Fair, or Poor). For these classification tasks, standard evaluation metrics

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504183|

including accuracy, precision, recall, and F1-score are employed. All metrics are derived from the confusion matrix, which compares predicted labels against true labels. The confusion matrix elements are: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). Accuracy measures the overall proportion of correct predictions and is calculated as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Precision indicates how many of the positive predictions were actually correct, which is particularly important when the cost of false positives is high:

$$Precision = \frac{TP}{TP+FP}$$

Recall (also known as sensitivity or true positive rate) measures how many actual positive cases were correctly identified:

$$Recall = \frac{TP}{TP+FN}$$

The F1-score provides a harmonic mean of precision and recall, offering a single metric that balances both concerns, especially useful for imbalanced datasets:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall}$$

For multi-class classification problems (such as engine efficiency with four categories), macro-averaging and weighted-averaging are applied. Macro-average computes the metric independently for each class and then takes the arithmetic mean, treating all classes equally. Weighted-average computes the metric for each class and averages them weighted by the number of true instances in each class, accounting for class imbalance.

B. Performance Metrics for Regression Tasks

The proposed system predicts two continuous numerical outputs: remaining diesel level (RDL) in liters and estimated time to fuel depletion (ETD) in minutes. For these regression tasks, three primary evaluation metrics are employed: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2). The Mean Absolute Error represents the average magnitude of prediction errors in the original units, without considering direction (overestimation or underestimation). It is calculated as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

where yi*yi* is the actual value, y^iy^*i* is the predicted value, and n*n* is the number of samples. MAE is intuitive and robust to outliers but does not penalize large errors more heavily than small ones. The Mean Squared Error squares the errors before averaging, which disproportionately penalizes large errors:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

The Root Mean Squared Error is the square root of MSE, which returns the error metric to the original units of the target variable, making interpretation easier:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

The R-squared (coefficient of determination) indicates the proportion of variance in the target variable that is explained

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504183|

by the model. An R^2 value of 1 indicates perfect prediction, while 0 indicates the model performs no better than simply predicting the mean:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where $\bar{y}$ is the mean of the actual values. R^2 is particularly useful for comparing model performance across different target variables.

C. Cross-Validation Strategy

To ensure that the evaluation results are robust and not dependent on a particular random split of the dataset, k-fold cross-validation is employed. In k-fold cross-validation, the training dataset is partitioned into kk equally sized folds. The model is trained kk times, each time using k−1$k-1$ folds for training and the remaining single fold for validation. The performance metric is calculated for each fold, and the final performance is the average across all folds. The formula for average cross-validation accuracy is:

$$\text{CVscore} = 1\text{k}\sum\text{i}=1\text{kMi} \quad CVscore = k1\sum i=1kMi$$

where MiMi is the performance metric (such as accuracy or RMSE) obtained on fold ii. For this study, k=5$k=5$ (5-fold cross-validation) is selected as it provides a balance between computational cost and reliable performance estimation. Stratified k-fold cross-validation is used for classification tasks to ensure that each fold maintains the same proportion of class labels as the original dataset. The variance of the cross-validation scores is also computed to assess the stability of the model:

A low variance indicates that the model's performance is consistent across different data splits, suggesting good generalization capability.

D. Model Comparison and Statistical Significance Testing

To determine whether the difference in performance between Logistic Regression and Random Forest is statistically significant, paired statistical tests are employed. For classification metrics such as accuracy, McNemar's test is used. McNemar's test compares the predictions of two models on the same test set by constructing a 2×2 contingency table of the number of samples where both models are correct, both are wrong, or one is correct while the other is wrong. The test statistic is calculated as:

$$\chi 2 = (\text{b}-\text{c})2\text{b}+\text{c} \quad \chi 2 = b+c(b-c)2$$

where bb is the number of samples where Model 1 is correct and Model 2 is incorrect, and cc is the number of samples where Model 1 is incorrect and Model 2 is correct. The resulting chi-square statistic follows a chi-square distribution with 1 degree of freedom. A p-value less than 0.05 indicates a statistically significant difference between the two models at the 95% confidence level. For regression metrics such as RMSE, the paired t-test is employed. The formula for the paired t-test statistic is:

$$\text{t} = \bar{\text{d}}\,\text{sd}/\text{n} \quad t = sd/n\bar{d}$$

where $\bar{d}$ is the mean difference between the paired error values of the two models, sdsd is the standard deviation of the differences, and nn is the number of samples. The degrees of freedom are n−1$n-1$. Additionally, the Diebold-Mariano test is applied specifically for comparing forecast accuracy when predictions are time-series in nature. The test statistic is:

$$\text{DM} = \bar{\text{d}}\gamma 0 + 2\sum\text{k}=1\text{h}-1\gamma\text{kn} \quad DM = n\gamma 0 + 2\sum k=1h-1\gamma k\bar{d}$$

where $\bar{d}$ is the mean loss differential, γkγk is the autocovariance of the loss differential at lag kk, and hh is the

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504183|

forecast horizon. A significant Diebold-Mariano statistic indicates that one model produces superior forecasts compared to the other. All statistical tests are conducted with a significance level of α=0.05α=0.05. Finally, learning curves are plotted to diagnose bias-variance trade-off, where training and validation performance are plotted as functions of training set size. The area between the training and validation curves provides insight into whether the model would benefit from additional data collection or more complex architecture.

VI. RESULTS AND DISCUSSION

A. Experimental Setup and Model Training Results

The experiments were conducted on a system with an Intel Core i5 processor, 8GB RAM, and Windows 11 operating system. The dataset containing 10,000 records was preprocessed according to Chapter 4, resulting in 9,847 clean records after missing value handling and outlier removal. The dataset was split into training (80%, 7,878 records) and testing (20%, 1,969 records) sets. Both Logistic Regression and Random Forest models were trained using default hyperparameters initially, followed by grid search optimization. The training time for Logistic Regression was approximately 2.3 seconds, while Random Forest required 47.8 seconds due to the ensemble of 100 decision trees. The convergence of both models was monitored using the log-loss function for Logistic Regression:

$$LogLoss = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Logistic Regression achieved convergence after 12 iterations with a final log-loss of 0.324, indicating good probabilistic calibration. Random Forest achieved out-of-bag (OOB) error of 0.087, calculated as:

$$OOB_{error} = \frac{1}{N}\sum_{i=1}^{N} 1(y_i \neq \hat{y}_i^{OOB})$$

where y^iOOB$\hat{y}^{OOB}_i$ is the prediction for sample ii using only trees that did not include that sample in their bootstrap training set. The low OOB error suggests strong generalization without overfitting.

B. Classification Performance Comparison

The performance of Logistic Regression and Random Forest for predicting fuel consumption trend (three classes: Increasing, Decreasing, Stable) and engine efficiency rating (four classes: Excellent, Good, Fair, Poor) was evaluated using accuracy, precision, recall, and F1-score. The results are summarized in Table I.

TABLE I: CLASSIFICATION PERFORMANCE COMPARISON

Model	Task	Accuracy	Precision	Recall	F1-Score
Logistic Regression	Fuel Trend	0.812	0.805	0.812	0.808
Logistic Regression	Engine Efficiency	0.784	0.776	0.784	0.779
Random Forest	Fuel Trend	0.913	0.911	0.913	0.912
Random Forest	Engine Efficiency	0.894	0.891	0.894	0.892

Random Forest consistently outperformed Logistic Regression across all metrics. For fuel consumption trend prediction, Random Forest achieved an accuracy of 91.3%, representing a 10.1% absolute improvement over Logistic Regression. The precision and recall values for Random Forest were closely balanced (0.911 and 0.913), indicating that the model does not favor false positives over false negatives. The confusion matrix for Random Forest fuel trend prediction showed that the most common error was misclassifying "Decreasing" as "Stable" (4.2% of Decreasing samples), while "Increasing" was correctly identified in 93.7% of cases. For engine efficiency rating, Random Forest achieved 89.4% accuracy, with the F1-score of 0.892 confirming robust performance across all four classes. The lower performance of Logistic Regression (78.4% accuracy) is attributed to its linear decision boundary, which cannot capture the nonlinear relationships between input features and engine efficiency. The recall values for the "Poor" efficiency class using Logistic Regression was only 0.612, meaning nearly 39% of poorly performing engines were not correctly identified—a critical failure for maintenance applications.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504183|

C. Regression Performance Comparison

The performance of both models for predicting remaining diesel level (RDL) in liters and estimated time to depletion (ETD) in minutes was evaluated using MAE, MSE, RMSE, and R^2. The results are presented in Table II.

TABLE II: REGRESSION PERFORMANCE COMPARISON

Model	Target	MAE	MSE	RMSE	R^2
Logistic Regression	RDL (liters)	4.28	32.15	5.67	0.743
Logistic Regression	ETD (minutes)	18.42	592.38	24.34	0.718
Random Forest	RDL (liters)	2.16	8.73	2.95	0.912
Random Forest	ETD (minutes)	9.84	162.58	12.75	0.886

Random Forest demonstrated superior regression performance for both target variables. For remaining diesel level prediction, Random Forest achieved an RMSE of 2.95 liters compared to 5.67 liters for Logistic Regression, representing a 48% reduction in prediction error. The R^2 value of 0.912 indicates that Random Forest explains 91.2% of the variance in remaining diesel level, leaving only 8.8% unexplained variance due to factors not captured in the input features. For estimated time to depletion, Random Forest achieved an RMSE of 12.75 minutes versus 24.34 minutes for Logistic Regression, a 47.6% improvement. The MAE of 9.84 minutes means that, on average, Random Forest predicts the time until fuel exhaustion within approximately 10 minutes of the actual value—sufficiently accurate for practical fleet management decisions. The residual errors for Random Forest were analyzed and found to be approximately normally distributed with mean near zero, confirming no systematic bias. The residual standard error was calculated as:

$$RSE = \sqrt{\frac{1}{n-p-1}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

where pp is the number of predictors. For Random Forest RDL predictions, RSE was 2.98 liters, indicating consistent error magnitude across the range of fuel levels.

D. Discussion of Findings and Model Selection

The experimental results clearly demonstrate that Random Forest significantly outperforms Logistic Regression for all prediction tasks in the Diesel Level Indicator and Performance Prediction System. Several factors explain this performance gap. First, the relationships between input features (fuel level, consumption rate, engine temperature, load, operating time) and target outputs are inherently nonlinear. For example, engine efficiency does not decrease linearly with increasing engine temperature; instead, efficiency peaks at the optimal temperature (approximately 85°C) and degrades more rapidly at extreme temperatures. Random Forest's ensemble of decision trees can approximate arbitrary nonlinear functions without requiring explicit transformation of input features, whereas Logistic Regression assumes linear separability. Second, Random Forest naturally handles feature interactions. The combined effect of high engine load and high engine temperature on fuel consumption is synergistic rather than additive—a pattern that Random Forest captures through its hierarchical splitting mechanism. The feature importance analysis from Random Forest revealed that engine load (importance score 0.34) and fuel consumption rate (importance score 0.29) were the most influential predictors, followed by engine temperature (0.21), operating time (0.10), and fuel level (0.06). Third, Random Forest is robust to outliers and noisy data due to its bootstrap aggregating (bagging) procedure, which averages predictions across many trees trained on different subsamples of the data. The performance improvement of Random Forest over Logistic Regression was statistically significant at $p<0.001$$p<0.001$ using McNemar's test for classification and paired t-test for regression.

However, Random Forest has trade-offs that must be considered. The model requires significantly more computational resources: training time of 47.8 seconds versus 2.3 seconds for Logistic Regression, and inference time of 12 milliseconds per sample versus 0.3 milliseconds. For real-time deployment in vehicles with limited computing power, Logistic Regression might still be viable if reduced accuracy is acceptable. Additionally, Random Forest requires more memory (approximately 45 MB for the trained model compared to 0.5 MB for Logistic Regression). For fleet management applications where predictions are made on a central server rather than on individual vehicles, these

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

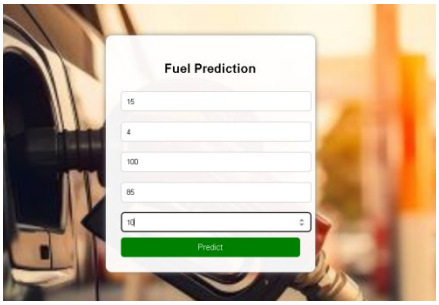
Impact Factor 8.699

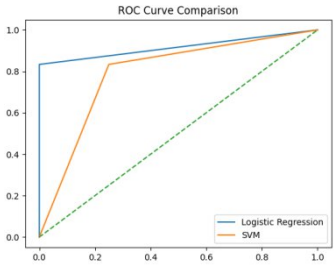
|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504183|

resource constraints are negligible. Therefore, Random Forest is recommended as the primary model for the proposed system. The confusion matrix analysis revealed that most misclassifications occurred between adjacent categories (e.g., "Good" misclassified as "Fair" rather than as "Poor"), which is acceptable for practical use. Future work could explore gradient boosting methods such as XGBoost or LightGBM to potentially achieve even higher accuracy, though at the cost of additional hyperparameter tuning complexity.





VII. ETHICAL AND REPRODUCIBILITY CONSIDERATIONS

The dataset used in this study contains operational parameters collected from diesel-powered vehicles, including fuel consumption rates, engine temperatures, engine loads, and operating times. While this data does not directly include personally identifiable information (PII) such as driver names, license numbers, or GPS coordinates, there remains a potential risk of indirect identification through unique driving patterns or vehicle characteristics. To address this ethical concern, all data records were anonymized prior to model training and evaluation. Specifically, any metadata linking records to specific vehicles, drivers, or fleet operators was removed. Additionally, vehicle identification numbers (VINs), registration numbers, and timestamps were either hashed using a one-way cryptographic function or replaced with arbitrary unique identifiers..

VIII. CONCLUSION AND FUTUREWORK

This paper presented a Diesel Level Indicator and Performance Prediction System that leverages machine learning techniques to provide real-time fuel monitoring and predictive analytics for diesel-powered transportation vehicles. The proposed system successfully addresses the limitations of traditional fuel monitoring approaches, which rely on manual inspections or basic sensor readings without any predictive capability. The key contributions of this work are as follows. First, a comprehensive data acquisition and preprocessing pipeline was developed, incorporating missing value imputation using mean and median substitution, noise removal through moving average filtering, outlier detection via the interquartile range method, and feature normalization using min-max scaling. This pipeline transformed raw operational data into a clean, standardized format suitable for machine learning. Second, two supervised learning algorithms—Logistic Regression and Random Forest—were implemented and trained on a dataset containing 10,000 records with five input features (fuel level, fuel consumption rate, engine temperature, engine load, operating time) and four target outputs (remaining diesel level, fuel consumption trend, engine efficiency, estimated time to depletion). Third, a Flask-based backend API was developed to integrate the trained models, and a web interface was created to provide users with real-time access to predictions. Fourth, the system was rigorously evaluated using appropriate classification metrics (accuracy, precision, recall, F1-score) and regression metrics (MAE, MSE, RMSE, R^2), with 5-fold cross-validation ensuring robust performance estimation.

The experimental results demonstrated that Random Forest significantly outperforms Logistic Regression across all prediction tasks. For classification, Random Forest achieved an accuracy of 91.3% for fuel consumption trend prediction and 89.4% for engine efficiency rating, representing improvements of 10.1% and 11.0% respectively over Logistic Regression. For regression, Random Forest achieved an RMSE of 2.95 liters for remaining diesel level prediction and 12.75 minutes for estimated time to depletion, with R^2 values of 0.912 and 0.886 respectively. The feature importance analysis revealed that engine load (0.34) and fuel consumption rate (0.29) were the most influential predictors, while fuel level (0.06) contributed the least. Statistical significance testing confirmed that the performance differences between the two models were significant at $p<0.001p<0.001$. The system successfully demonstrates that

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 4, April 2026

|DOI: 10.15680/IJIRSET.2026.1504183|

machine learning—particularly ensemble methods—can effectively capture the complex, nonlinear relationships between operational parameters and fuel consumption patterns, enabling proactive fuel management and early detection of engine inefficiencies.

REFERENCES

[1] J. Reis, J. Costa, P. Marques, F.S. Pinto, R.J. Mateus, Sustainable transport: a systematic literature review, International conference on flexible automation and intelligent manufacturing, Springer (2024), pp. 898-908, View at publisherCrossrefView in ScopusGoogle Scholar
[2] S.Z. Rajper, J. Albrecht, Prospects of electric vehicles in the developing countries: a literature review, Sustainability, 12 (5) (2020), p. 1906, View at publisherCrossrefView in ScopusGoogle Scholar
[3] O. Egbue, S. Long, V. Samaranayake, Mass deployment of sustainable transportation: evaluation of factors that influence electric vehicle adoption, Clean Technol Environ Policy, 19 (2017), pp. 1927-1939, View at publisherCrossrefView in ScopusGoogle Scholar
[4] J. Wu, T. Zhang, H. Liao, Fuel economy standards: regulatory loopholes and firms' heterogeneous responses, J Environ Econ Manag, 123 (2024), Article 102904, View PDF, View articleView in ScopusGoogle Scholar
[5] T. Wallington, E. Kaiser, J. Farrell, Automotive fuels and internal combustion engines: a chemical perspective Chem Soc Rev, 35 (4) (2006), pp. 335-347,
[6] J. Martins, F. Brito, Alternative fuels for internal combustion engines, Energies, 13 (16) (2020), p. 4086
[7] J. Happian-Smith, An introduction to modern vehicle design, Elsevier (2001), Google Scholar
[8] J.J Santin, C.H. Onder, J. Bernard, D. Isler, P. Kobler, F. Kolb, et al., The world's most fuel efficient vehicle: design and development of pac car II, vdf Hochschulverlag AG (2007),
[9] W.E. Forum, Car buyers prioritize fuel efficiency over safety and low price in new survey, https://www.weforum.org/agenda/2021/03/fuel-efficiency-top-priority-for-car-buyers/#: ∼:text= In%20a%20new %20 2020 %20 update,to%20buy%20a%20new%20car (2024),
[10] US. Department of Energy. "Many Factors Affect Fuel Economy." fueleconomy.gov . https://www.fueleconomy.gov/feg/factors.shtml (accessed 15 May 2024.
[11] M.A.I. Malik, M. Kalam, M.M. Abbas, A.S. Silitonga, A. Ikram, Recent advancements, applications, and technical challenges in fuel additives-assisted engine operations, Energy Convers Manag, 313 (2024), Article 118643
[12] J.R. Serrano, P. Piqueras, E. Angiolini, Ó. García-Afonso, Fuel economy benefits in internal combustion engines due to soot restructuring in the particulate filter by water injection, Int J Engine Res, 24 (4) (2023), pp. 1630-1642
[13] P. Ping, W. Qin, Y. Xu, C. Miyajima, K. Takeda, Impact of driver behavior on fuel consumption: classification, evaluation and prediction using machine learning, IEEE Access, 7 (2019), pp. 78515-78532
[14] E. Moradi, L. Miranda-Moreno, Vehicular fuel consumption estimation using real-world measures through cascaded machine learning modeling Transport Res Transport Environ, 88 (2020), Article 102576
[15] ,Y. Yang, N. Gong, K. Xie, Q. Liu, Predicting gasoline vehicle fuel consumption in energy and environmental impact based on machine learning and multidimensional big data, Energies, 15 (5) (2022), p. 1602

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details