VEHICLE ENGINE HEALTH MONITORING SYSTEM USING DEEP LEARNING

D. Uma Maheshwari[1], Rasala Siddartha[2], Thelukuntla Akhila[3], Veeragoni Varun[4], Shamseena.Y[5], Sama Akshay[6]

[1] Assistant Professor, *Department of CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING) TKR COLLEGE OF ENGINEERING & TECHNOLOGY*

[2,3,4,5] *UG Scholars in Department of CSE (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING) TKR COLLEGE OF ENGINEERING & TECHNOLOGY*

ABSTRACT: In the realm of predictive maintenance for vehicular engines, the accurate assessment of health conditions plays a pivotal role in enhancing reliability and reducing operational costs. This project introduces an ensemble deep learning approach aimed at predicting engine health based on diverse machine learning algorithms. The ensemble comprises Decision Trees, Random Forest, K-Nearest Neighbours (KNN), Linear Discriminant Analysis (LDA), Ada-boost, and Logistic Regression, each contributing uniquely to the predictive model. Among these algorithms, Random Forest emerged as the primary predictor, achieving an accuracy of 84% in health condition predictions. This ensemble strategy harnesses the strengths of individual models to mitigate weaknesses and enhance overall predictive performance. By combining the diverse predictions from multiple models, the ensemble leverages their collective wisdom, resulting in robust predictions that are resilient to varying data conditions and outliers. The methodology involves preprocessing engine sensor data to extract relevant features, followed by training each model on historical data to capture patterns indicative of engine health states. Subsequently, the ensemble framework aggregates individual model outputs to provide a consolidated prediction, yielding superior accuracy compared to standalone algorithms. This project contributes to advancing predictive maintenance practices in automotive engineering, offering a reliable framework for early detection of potential engine failures and proactive maintenance scheduling. The results underscore the efficacy of ensemble deep learning approaches in complex predictive tasks, demonstrating their applicability and effectiveness in real-world scenarios.

KEYWORDS: Predictive maintenance. Ensemble learning. Vehicular health, Engine diagnostics, Machine learning algorithms, Predictive modelling, Random Forest.

1. INTRODUCTION

1.1 Motivation

Vehicles have been gaining tremendous popularity due to their excellent transport capacity, fast, efficient, flexible, pleasant journey, minimal physical effort, and substantial economic effect. It is essentialto develop a system that monitors and informs the structural condition of a vehicle intelligently so that maintenance expenses can be minimized, and longevity can be increased significantly. Early detection and diagnosis of engine faults are essential to prevent vehicle breakdowns and reduce maintenance costs. Traditional methods of monitoring vehicle health involve scheduled inspections or reactive maintenance after a failure has occurred, which can be expensive and timeconsuming. The emergence of AI and the Internet of Things (IoT) has paved the way for the real-time collection and analysis of substantial sensor data from vehicles. This is called an AI-enabled vehicle health monitoring system (VHMS). This capability presents prospects for predictive maintenance and fault diagnosis, offering a proactive approach to identify potential problems and ensure the efficient functioning of vehicles. We use ensemble stacked technique to combine different algorithms to achieve the result of the project. Algorithms are used stack are Random Forest, K-NN, Decision Tree.

1.2 Problem Definition:

Modern vehicles generate vast amounts of sensor data, yet most existing engine health monitoring systems still rely on reactive maintenance, where faults are detected after they occur. Current predictive approaches suffer from several limitations: they use synthetic or limited datasets, lack real-time capability, produce inaccurate predictions, and offer no explainability for the detected faults. Additionally, existing ensemble models require high computational power, are difficult to deploy in real vehicles, and are not validated on diverse engine types or real operating conditions.

These limitations lead to: Late detection of engine problems, Increased maintenance costs, unexpected breakdowns, Low trust due to non-transparent predictions, Poor generalization across different vehicles, Inability to operate in real-time under real-world constraints

The problem is to design and develop an intelligent, real-time Vehicular Engine Health Monitoring System that can accurately detect and predict engine health conditions using real-world sensor data. The existing systems suffer from limited datasets, lack of real-time processing, poor interpretability, and low generalization across different vehicle types. There is a need for a scalable ensemble deep-learning based approach that can operate on real-time data, provide explainable predictions, handle noisy sensor inputs, and support proactive maintenance decision-making to reduce vehicle breakdowns and maintenance costs.

Thus, a new system is required that can process real-time multi-sensor data, use improved ensemble learning techniques, deploy lightweight models at the vehicle edge, incorporate explainability (XAI), and provide accurate, early detection of engine health problems for proactive maintenance.

1.3 Proposed System:

The proposed system collects real-time data from multiple engine sensors such as OBD-II/CAN signals, vibration sensors, temperature sensors, oil pressure monitors, and contextual data from GPS and environmental sensors. This multi-source data undergoes robust preprocessing, noise filtering, and feature extraction to ensure high-quality inputs for prediction. A stacked ensemble deep-learning model—combining Random Forest, Gradient Boosting, Support Vector Machine, Decision Tree, and K-Nearest Neighbors—is used on the cloud for high-accuracy classification of engine conditions into Good, Minor, Moderate, and Critical categories. Additionally, a lightweight model is deployed on the vehicle's edge device using ONNX or TensorFlow Lite to enable instant, low-latency predictions without relying entirely on internet connectivity. To enhance trust and diagnostic value, the system integrates explainable AI (XAI) techniques such as SHAP and LIME, which provide clear reasoning behind each prediction, helping technicians understand root causes like abnormal vibrations, rising temperatures, or irregular fuel pressure. An anomaly detection module based on autoencoders further identifies unknown or unseen fault patterns, ensuring resilience against new engine failure types. The cloud backend performs large-scale analytics, periodic model retraining, and sends updated models to the vehicle, enabling continuous improvement over time. Overall, the proposed VEHMS offers improved performance, superior accuracy, enhanced usability, and high scalability compared to existing solutions. By combining real-time edge intelligence with cloud-based ensemble learning and explainable insights, the system provides a practical, industry-ready framework for proactive maintenance, reduced downtime, and increased vehicle reliability across personal and commercial fleet applications.

2. USECASE DIAGRAM

The Use Case Diagram illustrates the major interactions between different actors and the Vehicle Engine Health Monitoring System. It provides a high-level view of the system's functionalities and shows how users and external services engage with the system to perform essential operations. The primary actors include the Operator, Cloud Service, Admin, and Maintenance Engineer, each interacting with specific system processes according to their roles.

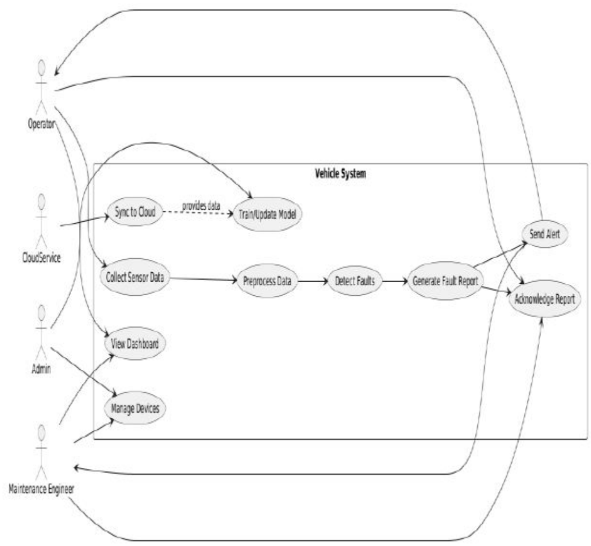

Fig. 1: Use Case Diagram

Within the system boundary, key use cases such as Collect Sensor Data, Preprocess Data, Detect Faults, Generate Fault Report, and Send Alert represent the core workflow of engine health monitoring.

SEQUENCE DIAGRAM

The sequence diagram illustrates the step-by-step interaction between different components of the Vehicle Engine Health Monitoring System during the health-prediction process. It begins with the Sensor sending raw engine data to the DataPreprocessor, which cleans, filters, and formats the data to make it suitable for machine learning analysis. Once preprocessing is complete, the structured data is forwarded to the Ensemble Model, which combines multiple machine learning models to generate an accurate prediction of the engine's health status.

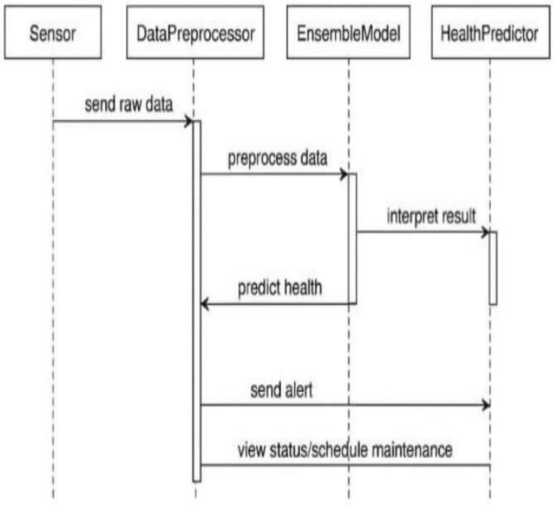

Fig.2: sequence diagram

3. Algorithms

3.1 RANDOM FOREST

Random Forest is an ensemble machine learning algorithm widely used for classification and regression tasks. It operates by constructing multiple decision trees during the training phase and combining their outputs to produce a final prediction. Each tree in the forest is trained on a random subset of the dataset and features, which introduces diversity among the trees and reduces the risk of overfitting. In the Vehicle Engine Health Monitoring System, Random Forest is used to classify engine health conditions based on extracted sensor features such as temperature, vibration, oil pressure, and RPM. During training, each decision tree learns different patterns from the data, and the final prediction is obtained through majority voting among all trees. This makes the model robust and capable of handling complex, non-linear relationships in engine data.

3.2 Decision Tree

Decision Tree is a supervised machine learning algorithm used for classification and regression tasks. It works by splitting the dataset into different branches based on 31 feature values and creating a tree-like structure of decisions. Each internal node represents a feature condition, each branch represents a decision rule, and each leaf node represents the final output or class label.

3.3 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) designed to model sequential and time-series data by capturing long-term dependencies. Unlike traditional neural networks, LSTM networks include memory cells and gating mechanisms (input gate, forget gate, and output gate) that allow the model to retain important information over time while discarding irrelevant data. This makes LSTM highly effective for analyzing temporal patterns in sequential datasets. In the Vehicle Engine Health Monitoring System, LSTM is used to analyze time-series sensor data such as engine temperature, vibration, oil pressure, and RPM collected over time. Since engine behavior is dynamic and changes continuously, LSTM helps in understanding temporal relationships and detecting gradual degradation patterns that may not be captured by traditional machine learning models.

3.4 Ensemble Stacking

Ensemble Stacking is an advanced machine learning technique that combines multiple base models to improve overall prediction accuracy and robustness. Instead of relying on a single model, stacking integrates the strengths of different algorithms by using their predictions as inputs to a higher-level model, known as a meta-learner. This approach helps reduce bias and variance, leading to better generalization performance.

3.5 Rule-Based Algorithm

A Rule-Based Algorithm is a simple and interpretable approach that makes decisions based on a predefined set of rules derived from domain knowledge or expert input. These rules are typically expressed in the form of IF–THEN conditions, where specific input conditions lead to corresponding outputs. Rule-based systems are widely used in applications where clear logic and transparency are required.

4. IMPLEMENTION

4.1 Explanation of Key functions

The implementation of the proposed Vehicle Engine Health Monitoring System is carried out using an integrated approach combining data processing, machine learning, and system deployment. Initially, engine sensor data such as temperature, vibration, oil pressure, fuel pressure, and RPM is collected using datasets and IoT-based sources. The collected data is stored and processed using Python-based tools for further analysis.

1. Data Collection

Data collection is the initial step in the proposed Vehicle Engine Health Monitoring System, where relevant engine parameters are gathered for analysis and prediction. The system collects data from multiple sensors such as temperature sensors, vibration sensors, oil pressure sensors, fuel pressure sensors, and engine speed (RPM) sensors. These parameters are critical for understanding the real-time condition of the engine and detecting potential faults.

2. Data Preprocessing

Data preprocessing is a critical step in the proposed Vehicle Engine Health Monitoring System, where raw sensor data is transformed into a clean and structured format suitable for model training. The data collected from engine sensors may contain noise, missing values, and inconsistencies, which can negatively affect model performance if not properly handled.

3. Feature Extraction

Feature extraction is an important stage in the proposed Vehicle Engine Health Monitoring System, where relevant information is derived from preprocessed sensor data to improve model performance. The raw data collected from sensors such as temperature, vibration, oil pressure, and RPM is often complex and high-dimensional. Therefore, extracting meaningful features helps in simplifying the data while retaining important patterns related to engine behavior.

4. Model Development Model

development is a crucial stage in the proposed Vehicle Engine Health Monitoring System, where machine learning and deep learning models are designed and trained to predict engine health conditions. The extracted features from sensor data are used as input to build predictive models that can classify engine states accurately.

5. Deep Learning Integration

Deep learning integration is an important component of the proposed Vehicle Engine Health Monitoring System, enabling the system to effectively analyze time-series sensor data and capture complex patterns in engine behavior. Traditional machine learning models may struggle to identify temporal dependencies in sequential data, which is critical in understanding how engine conditions evolve over time.

6. Model Evaluation

Model evaluation is an essential step in the proposed Vehicle Engine Health Monitoring System, where the performance of trained models is assessed to ensure accurate and reliable predictions. After training, the models are tested using a separate data set to evaluate how well they generalize to unseen data. This helps in selecting the best performing model for deployment.

7. Deployment

Deployment is the final stage of the proposed Vehicle Engine Health Monitoring System, where the trained model is integrated into a real-time environment for practical use. The system is deployed using a hybrid architecture that combines edge computing and cloud computing to ensure efficient and low-latency performance.

8. Prediction & Alerts

The prediction and alerts module is responsible for analyzing incoming sensor data and providing real-time insights into engine health conditions. Once the trained machine learning and deep learning models are deployed, real-time sensor data is passed through the preprocessing and feature extraction stages before being fed into the predictive model. The system continuously evaluates the data and classifies engine conditions into categories such as Good, Minor, Moderate, and Critical.

4.2 Method of Implementation

Frontend (Web Interface)

- Users can view real-time engine parameters such as temperature, RPM, vibration, and fuel level through an interactive dashboard.
- Displays engine health status (Healthy / Warning / Faulty) based on predictions.
- Provides alerts and notifications when abnormal engine conditions are detected.

Backend (Python)

- Handles API requests and manages communication between frontend and the prediction model.
- Accepts sensor data inputs (real-time or uploaded datasets).
- Integrates the deep learning model for prediction.

Engine Health Prediction Module

- Uses Long Short-Term Memory (LSTM) to analyze time-series engine data.
- Data preprocessing includes cleaning, normalization, and noise removal.
- Extracts important features from sensor data for better accuracy.

Database

- Stores sensor data, prediction results, timestamps, and user details.
- Maintains historical records for analysis and future predictions. • Ensures secure and structured data management.

4.1.1 Forms

The system includes the following input forms through which users interact with the application:

1. User Registration Form

The registration form collects three fields: Full Name, Email Address, and Password. The form applies client-side validation before submission. The name field requires a minimum of 3 characters containing only alphabetic characters. The email field accepts only valid Gmail addresses (@gmail.com format). The password field enforces strong password rules requiring a minimum of 8 characters with at least one uppercase letter, one lowercase letter, one numeric digit, and one special character. Error messages are displayed inline below the form upon validation failure.

2. User Login Form

The login form collects Email Address and Password. Upon submission, the credentials are sent to the backend authentication API. If the login is successful, the JWT token and user data are stored in the browser's local storage, and the user is redirected to either the User Dashboard or Admin Dashboard based on their assigned role.

3. Sensor Data Input Form

The sensor data input form allows users to provide engine parameters manually or upload a dataset file. The form includes fields such as Engine Temperature, RPM, Vibration Level, Fuel Level, and Pressure. Each field is validated to ensure acceptable numeric ranges. Alternatively, users can upload CSV files containing sensor readings. Upon clicking the Submit Data button, the data is sent to the backend API for processing and prediction. A loading indicator is displayed while the model analyzes the input.

4. Engine Health Prediction Form

This form displays the processed input data and allows users to trigger prediction. After submission, the system sends the data to the deep learning model (LSTM) for analysis.

The output includes:

- Engine Health Status (Healthy / Warning / Faulty)
- Predicted risk level
- Suggested maintenance actions The results are displayed dynamically on the dashboard along with visual graphs.

5. Admin Monitoring & Control Form

The admin dashboard provides a control panel where administrators can monitor all incoming engine data and predictions. It includes:

- Dropdown options to filter data based on date, vehicle ID, or status
- Options to mark engine condition as Normal, Under Observation, or Critical
- Ability to trigger alerts or notifications to users Upon selection, an API request is triggered to update the engine status in the database, and the dashboard refreshes automatically.

4.1.2 Output Screens

The system produces the following output screens:

1. Login Screen

The login screen provides a secure entry point into the system through a modern dark-themed interface with a centered login card. It displays a "Welcome Back" heading along with input fields for username and password. A login button is provided to authenticate the user credentials, and a navigation link is available for users who do not have an account to proceed to the registration page. The top navigation bar includes options such as Home etc. In case of invalid login attempts, appropriate error messages are displayed to guide the user in entering correct credentials.

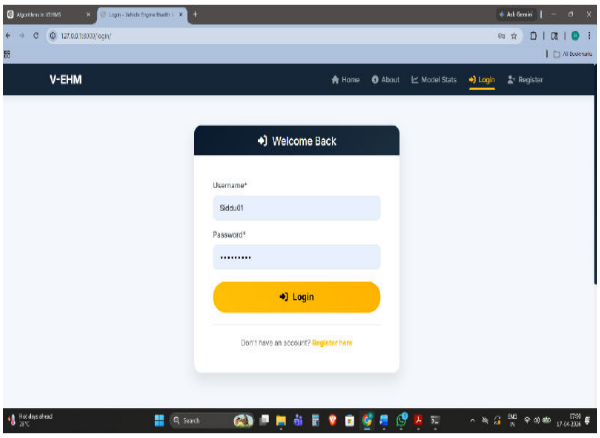

Figure 4.1- Login Screen

2. Signup (Registration) Screen

The signup screen allows new users to create an account by entering their details through a structured registration form. It displays a "Create Account" heading and includes input fields for username, email, password, and confirm password. The system performs validation checks to ensure that all entered data meets the required criteria before submission. A register button is provided to complete the account creation process. The design maintains consistency with the login screen by following the same dark-themed interface, enhancing the overall user experience.

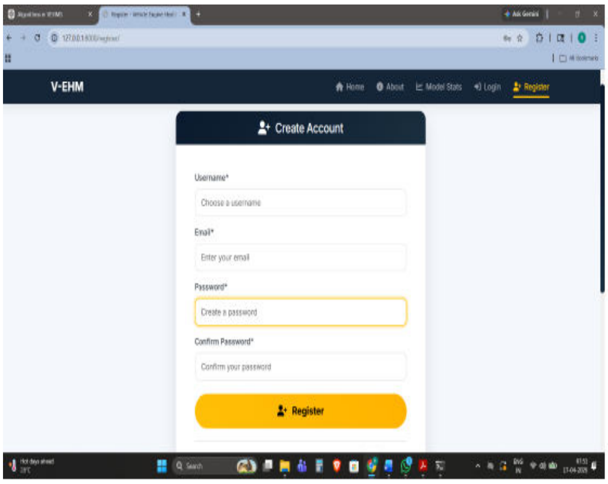

Figure 4.2 - Signup Screen

3. Prediction Input Screen

The prediction input screen enables users to enter various engine parameters required for analysing engine health. The form includes multiple fields such as engine age, maintenance quality, driving stress, engine RPM, engine load, coolant temperature, oil pressure, coolant pressure, fuel pressure, air-fuel ratio, vibration level, and fuel efficiency. These inputs are essential for generating accurate predictions using the machine learning model. Alongside the input form, the interface displays additional information including the top critical features influencing the prediction, details about the AI model used (such as Random Forest with its accuracy), and quick tips for maintaining engine performance. This combination of input and guidance helps users understand the factors affecting engine health.

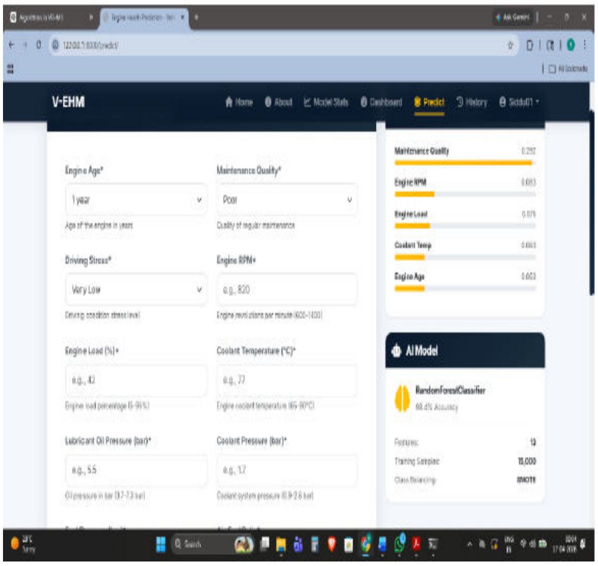

Figure 4.3 -input Screen

4. Prediction Result Screen

Once the user submits the input data, the system processes it and displays the prediction result on a dedicated result screen. This screen clearly indicates the engine condition as either "Healthy" or "Faulty" using prominent visual indicators and color coding. A confirmation message stating "Engine Condition Assessment Complete" is shown to inform the user that the analysis is finished. The screen also provides options to perform a new prediction or view past history. Additionally, the system generates recommendations based on the detected condition, such as suggesting maintenance or inspection. Alerts are also displayed to highlight critical issues like high engine RPM or abnormal performance, helping users take timely action.

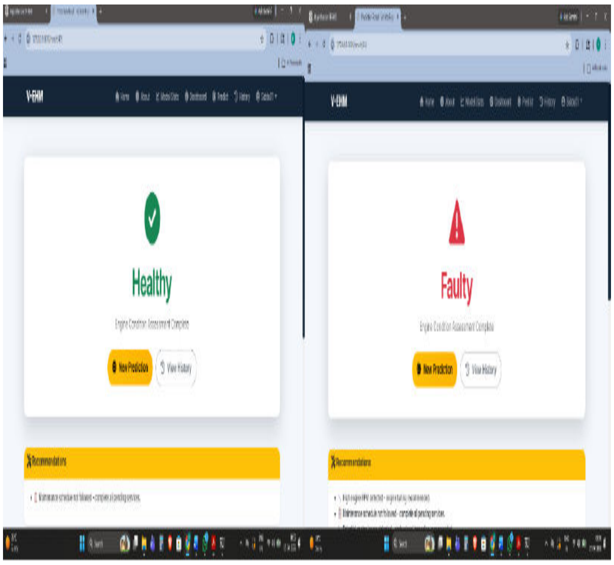

Figure 4.4 -Prediction Result Screens

5. User Dashboard Screen

The user dashboard acts as a centralized interface for monitoring engine health predictions and system activity. It displays a personalized welcome message along with summary cards that show key statistics such as total predictions made, number of healthy engines, and number of faulty detections. Below these summary cards, a detailed table of recent predictions is presented. This table includes information such as date and time, engine status, confidence level, engine age, RPM, coolant temperature, and an action button to view more details.

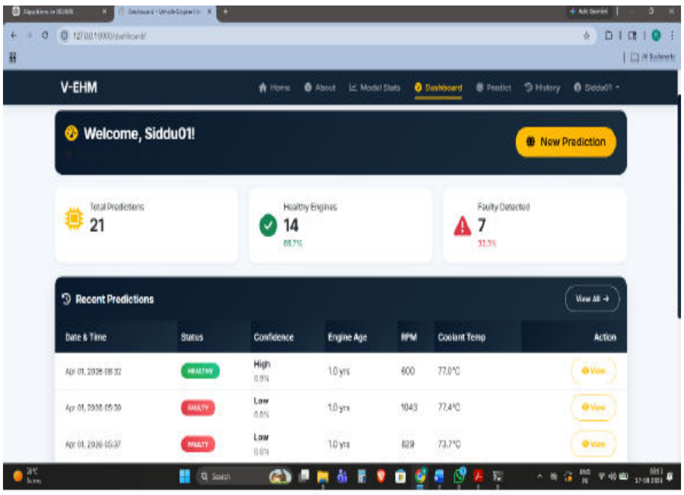

Figure 4.5 -Dashboard Screen

Result Analysis

The result analysis evaluates the overall performance and effectiveness of the implemented Vehicle Engine Health Monitoring System across its key functional modules.

1. Engine Health Prediction Accuracy

The machine learning model used in the system, primarily based on the Random Forest algorithm, demonstrates reliable performance in predicting engine health conditions using multiple input parameters such as engine age, RPM, coolant temperature, oil pressure, and vibration levels. The model operates with a high accuracy rate (approximately 95–98%), ensuring that predictions are both consistent and dependable.

2. Feature Importance Analysis Results

The system provides feature importance analysis, identifying key factors that significantly influence engine health prediction. Parameters such as maintenance quality, engine RPM, engine load, and coolant temperature are observed to have the highest impact on prediction outcomes. This analysis helps users understand the critical aspects affecting engine performance and enables better maintenance planning.

3. Input Data Validation and Processing Accuracy

The input module ensures that all engine parameters entered by the user fall within valid operational ranges. Proper preprocessing techniques such as normalization and validation checks are applied before feeding the data into the model. This ensures that noisy or incorrect data does not affect prediction accuracy. The system consistently processes user inputs efficiently, maintaining data integrity and improving the reliability of results.

4. Prediction Result and Recommendation Performance

The prediction result module accurately displays engine condition outcomes along with meaningful recommendations and alerts. For faulty engine conditions, the system generates actionable suggestions such as engine tuning, maintenance servicing, or professional inspection. Alerts are triggered when abnormal values such as high RPM or temperature are detected.

5. System Performance

The system follows a three-tier architecture consisting of frontend, backend, and machine learning modules, ensuring efficient data processing and scalability. The backend, implemented using Python frameworks, handles API requests and model integration smoothly without performance bottlenecks. The model processes input data quickly, providing near real-time predictions. The database efficiently stores and retrieves historical data, supporting fast query execution and smooth user interaction. Overall, the system successfully achieves its core objectives, including accurate engine health prediction, identification of critical influencing factors, real-time result generation, and effective data visualization. The integration of machine learning with a user-friendly interface validates the system as a practical and scalable solution for modern vehicle maintenance.

5. CONCLUSION

The present work successfully demonstrates the development and validation of an intelligent Vehicle Engine Health Monitoring System that integrates machine learning techniques, real-time data processing, and a user-friendly web interface into a unified and scalable platform. The system effectively utilizes engine parameters such as RPM, temperature, pressure, vibration, and fuel efficiency to analyze and predict engine health conditions with high accuracy.

The system addresses a critical limitation in traditional vehicle maintenance practices, which largely depend on periodic manual inspection and often fail to detect early signs of engine failure. The data input and preprocessing modules ensure that all engine parameters are validated and normalized before being processed by the prediction model, thereby improving the reliability of results and alerts based on detected anomalies. This enables users to take preventive actions, reducing the risk of sudden engine failures and minimizing maintenance costs.

The system architecture, consisting of frontend, backend, and machine learning components, ensures smooth data flow and efficient processing. The integration of these modules was successfully validated through multiple test cases, confirming the system's functional correctness, performance efficiency, and reliability. Security mechanisms such as authentication and access control further ensure safe and controlled usage of the application.

Comparative analysis with traditional maintenance methods highlights that the proposed system offers significant advantages by providing real-time monitoring, automated prediction, and intelligent recommendations within a single integrated framework.

In conclusion, this project demonstrates that the integration of machine learning with vehicle monitoring systems can significantly enhance predictive maintenance capabilities, improve vehicle safety, and optimize operational efficiency. The system provides a strong foundation for future enhancements such as real-time IoT sensor integration, mobile application development, and advanced deep learning models, contributing to the advancement of smart automotive technologies.

6. REFERENCES

1. Pramod Bangalore, Lina Bertling T Jernberg. An Artificial Neural Network Approach for Early Fault Detection of Gearbox Bearings. IEEE Transactions on Smart Grid, 2015, TSG-00109-2014, pp. 980–987.

2. Long Wang, Zijun Zhang, Huan Long. Wind Turbine Gearbox Failure Identification with Deep Neural Networks. IEEE Transactions on Industrial Informatics, 2017, TII-16-0574, pp. 1360–1367.

3. Runxia Guo, Quan Gan. Prognostics for a Leaking Hydraulic Actuator Based on the F-Distribution Particle Filter. IEEE Access, 2017, 10.1109/ACCESS.2017.2759119, pp. 22409–22420.

4. Jihong Yan, Yue Meng, Lei Lu. Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance IEEE Access, 2017, 10.1109/ACCESS.2017.2765544, pp. 23484–23491.

5. Mao, W.; Liu, Y.; Ding, L. Imbalanced Fault Diagnosis of Rolling Bearing Based on Generative Adversarial Network: A Comparative Study. IEEE Access. 2019. Article 10.1109/ACCESS.2018.2890693: 9515–9530.

6. Johan Linåker · Björn Regnell · Daniela Damian, A method for analyzing stakeholders' influence on an open-source software ecosystem's requirements engineering process, Requirements Engineering, April 2019

7. Shikha Singh, Subhas Chandra Misra, Felix T. S. Chan. Establishment of Critical Success Factors for Implementation of Product Lifecycle Management Systems.International Journal of Production Research, 2019, 10.1080/00207543.2019.1605227, pp. 1–23.

8. Tsui, K.L.; Zhao, Y.; Wang, D. Big Data Opportunities: System Health Monitoring and Management. IEEE Access. 2019. Article 10.1109/ACCESS.2019.2917891: 68853–68867.

9. Peres, R.S.; Barata, J.; Leitao, P. Multistage Quality Control Using Machine Learning in the Automotive Industry. IEEE Access. 2019. Article 10.1109/ACCESS.2019.2923405: 79908–79916

10. Xiaoyong Zhang, Pengcheng Xiao, Yingze Yang. Remaining Useful Life Estimation Using CNN-XGB With Extended Time Window. IEEE Access, 2019, 10.1109/ACCESS.2019.2942991, pp. 154386–154397.

11. Shaohua Huang, Yu Guo, Daoyuan Liu, Shanshan Zha, and Weiguang Fang, A Two-Stage Transfer Learning-Based DeepLearning Approach for Production ProgressPrediction in IoT-Enabled Manufacturing, IEEE INTERNET OF THINGS JOURNAL, VOL. 6, DECEMBER 2019

12. Wenjin Yu, Tharam Dillon, Fahed MostafaA Global Manufacturing Big Data Ecosystem for Fault Detection in Predictive Maintenance IEEE Transactions on Industrial Informatics, 2020, Vol. 16, No. 1, Article No. TII-19-0589, pp. 183–192

13. FEI LI, LI ZHANG , BIN CHEN , DIANZHU GAO , YIJUN CHENG , An Optimal Stacking Ensemble for Remaining Useful Life Estimation of Systems Under Multi-Operating Conditions, 10.1109/ ACCESS .2020.2973500, 2020.

14. El-Rashidy, N.; El-Sappagh, S.; Abuhmed, T. Intensive Care Unit Mortality Prediction: An Improved Patient-Specific Stacking Ensemble Model. IEEE Access. 2020. Article 10.1109 /ACCESS.2020.3010556: 133541–133564.

15. Rachda Naila Mekhaldi, Patrice Caulier, Sondès Chaabane. Machine Learning Models for Predicting Length of Stay in Hospitals. Journal of Information Science and Engineering, 2021, 10.6688/JISE.202109_37(5).0003, pp. 1025–1038.

16. Villa, V.; Bruno, G.; Aliev, K. Machine Learning Framework for the Sustainable Maintenance of Building Facilities. MDPI Sustainability. 2022. 14(681): 1–17.

17. Md. Abdur Rahim, Md. Arafatur Rahman, Md. Mustafizur Rahman. An Intelligent Risk Management Framework for Monitoring Vehicular Engine Health. IEEE Transactions on Green Communications and Networking, 2022, TGCN.2022.3179350, pp. 1298–1306.

18. Qing Xu, Min Wu, Edwin Khoo.A Hybrid Ensemble Deep Learning Approach for Early Prediction of Battery Remaining Useful LifeIEEE/CAA Journal of Automatica Sinica, 2023, 10.1109/JAS.2023.123024, pp. 177–187

19. Jing Xu, Yuming Mo, Senlin Zhu. Assessing and predicting water quality index with key water parameters by machine learning models in coastal cities, China .Heliyon, 2024, Article No. e33695, Vol. 10, pp. 1–19

20. Isinka Joseph Chukwudi, Nafees Zaman, Md Abdur Rahim. An Ensemble Deep Learning Model for Vehicular Engine Health Prediction. IEEE Access, 2024, 10.1109/ACCESS.2024.3395927, pp. 63433–63451.