RV College of Engineering
Experiential Learning Report
Project-Based Learning

2024-25

Title of the Project

AI and ML techniques for real-time diagnostics and fault detection in automotive systems

Student(s)

USN	Name
1RV22CS143	Pranav Darshan
1RV22CS154	Raghuveer Rajesh
1RV22IS078	Vijeth G
1RV22IS027	Jeevith P
1RV22CY046	Prathica Shetty M

Mentor

Dr. Mohana
Assistant Professor
Department of Computer Science and Engineering

1. Introduction

The increasing complexity of modern automotive systems has necessitated the use of intelligent fault detection mechanisms to ensure vehicle reliability and safety. Traditional maintenance methods are often reactive, leading to unexpected breakdowns and costly repairs. AI and ML techniques enable real-time monitoring, predictive maintenance, and proactive fault diagnostics, minimizing vehicle downtime and optimizing performance.

Machine learning models analyze sensor data from vehicles in real time, detecting anomalies and predicting component failures before they occur. This transition from reactive to predictive maintenance significantly improves efficiency in automotive diagnostics. The integration of AI in vehicle health monitoring not only enhances operational safety but also reduces maintenance costs by preemptively identifying potential faults.

Additionally, the rise of IoT and connected vehicles has further amplified the importance of real-time diagnostics. By continuously analyzing critical parameters such as engine temperature, oil pressure, and battery voltage, AI-based systems can provide early warnings and suggest corrective actions. These innovations contribute to an overall improvement in vehicle lifespan and reliability, benefiting individual owners, fleet operators, and manufacturers alike.

2. Problem Definition

2.1. Problem Statement:

The automotive industry lacks an intelligent predictive maintenance system capable of real-time monitoring and early warning of potential failures. This results in inefficiencies, unexpected breakdowns, and increased operational costs. A robust AI-based system is required to predict component failures in advance and improve maintenance schedules.

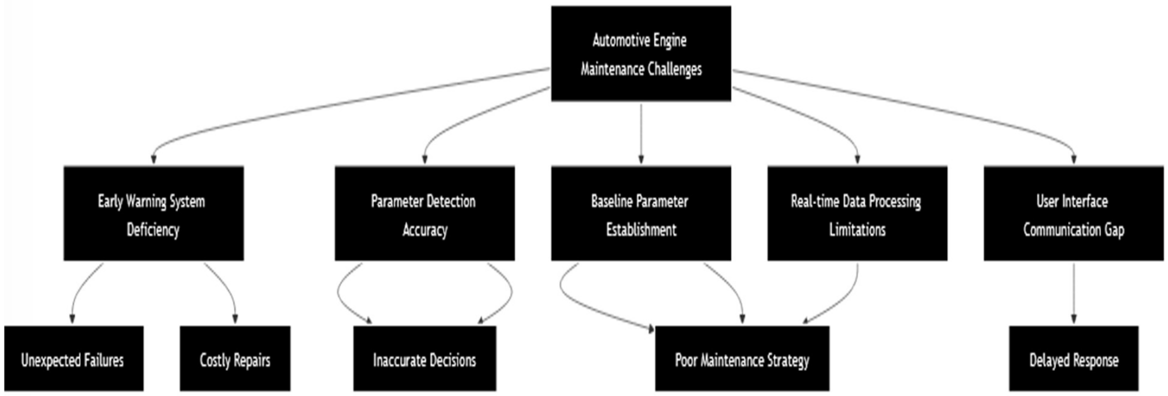

Fig1: Understanding the problem

2.2. Background Information: (literature review)

Automobiles consist of various interconnected mechanical and electronic components that require routine inspection and maintenance. Traditional fault detection methods depend heavily on manual inspections, periodic maintenance, or rule-based systems embedded in vehicle control units. These conventional approaches are often inefficient in identifying faults before they escalate into major failures.

With advancements in machine learning, data-driven predictive maintenance has gained prominence. By leveraging historical and real-time sensor data, AI models can identify patterns and anomalies that signal potential faults. Several studies have demonstrated the effectiveness of ML techniques such as Support Vector Machines (SVM), Decision Trees, and Neural Networks in automotive diagnostics. However, these models often require extensive feature engineering and may not generalize well across different vehicle types.

The adoption of Gradient Boosting Machines (GBM) has addressed some of these challenges by offering better predictive accuracy and handling non-linear dependencies in complex automotive systems. Additionally, the integration of Internet of Things (IoT) devices in modern vehicles has enabled real-time data collection, further enhancing predictive capabilities. These AI-driven systems help in mitigating risks associated with unexpected breakdowns, ensuring timely maintenance interventions, and ultimately extending vehicle lifespan.

Furthermore, regulatory bodies and automotive manufacturers are increasingly focusing on predictive maintenance to improve road safety and compliance with emission norms. Advanced fault detection mechanisms contribute to reducing hazardous failures, minimizing environmental impact, and optimizing resource utilization in vehicle maintenance operations.

3. Objectives

3.1. Primary Objectives:

- Develop an intelligent predictive maintenance system using AI and ML.
- Monitor vehicle sensor data in real time to detect faults early.
- Provide accurate predictions on component degradation and failure probabilities.
- Develop a user-friendly web interface for real-time monitoring and predictions.

3.2. Secondary Objectives:

- Enhance vehicle reliability and safety.
- Optimize maintenance schedules to minimize downtime and costs.
- Improve system scalability to support various vehicle models and sensor configurations.
- Facilitate seamless integration with existing vehicle diagnostic tools and fleet management systems

4. Methodology

4.1 Approach:

The system follows a data driven approach, integrating real-time vehicle sensor data with an ML-based fault detection model. The workflow includes:

- **Data Collection:** Capturing sensor data (temperature, RPM, pressure, etc.).
- **Preprocessing:** Cleaning, normalization, and feature engineering.
- **Model Training:** Implementing a TabNet model for predictive maintenance.
- **Web Interface:** Developing a React-based dashboard for visualization.
- **Backend Integration:** Utilizing Spring Boot to handle API requests and manage data flow.
- **LLM Integration:** Passing sensor readings to a large language model (LLM), such as LLaMA, for deeper analysis and enhanced recommendations.

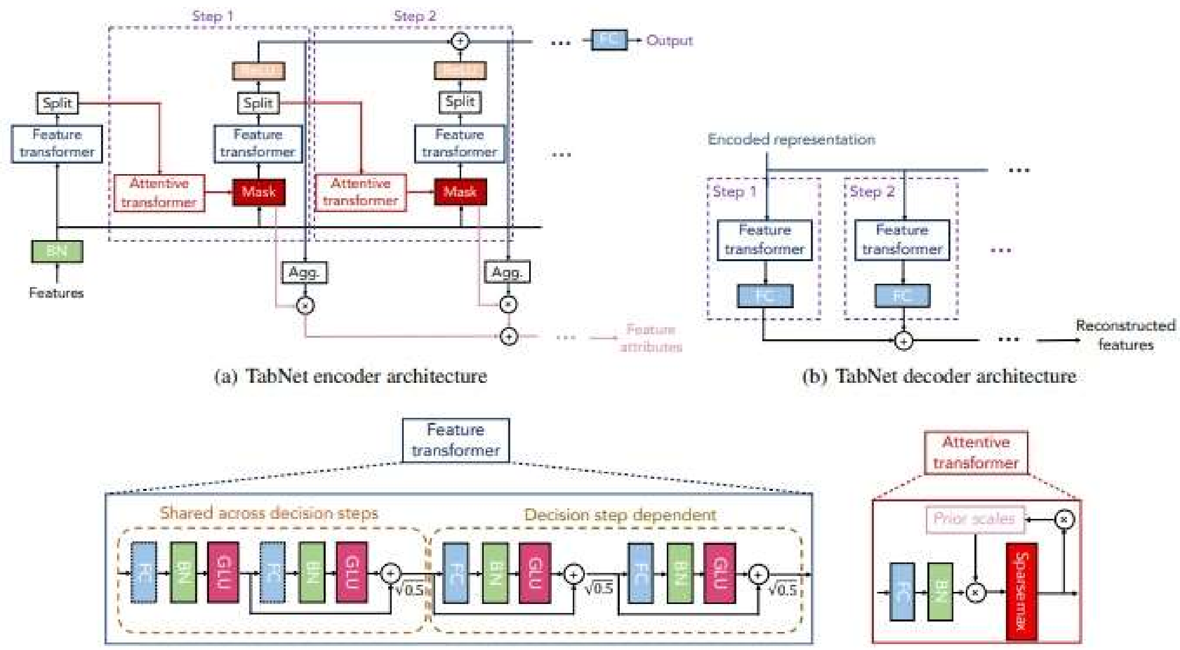

Fig 2: ROC curve

4.2 Procedures:

- Data acquisition from vehicle sensors.
- Preprocessing for consistency and outlier detection.
- TabNet model training with hyperparameter tuning.
- Integration of the trained model into a web-based system.
- Backend implementation using Spring Boot for API services.
- Passing sensor readings to an LLM (e.g., LLaMA) to analyze vehicle health and predict breakdown timelines.
- Deployment and real-time monitoring.

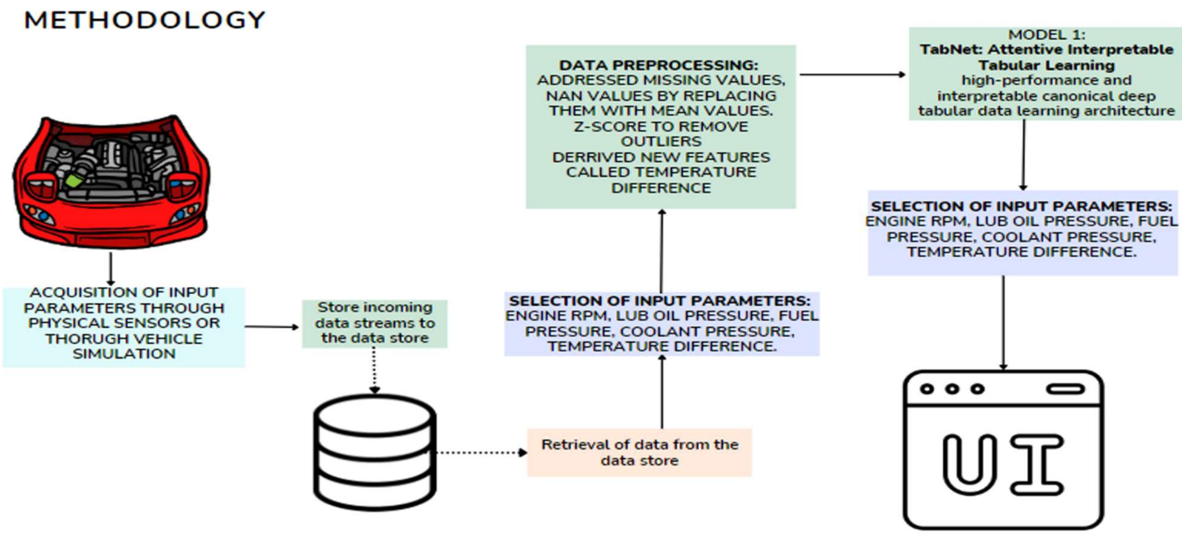

Fig 3: Process Flow

5. Project Execution

5.1 Planning and Design:

The planning and design phase involved extensive research and collaboration to define project goals, data sources, and system architecture. Initially, brainstorming sessions were conducted to evaluate different machine learning models and data processing techniques. The dataset selection was a crucial step, requiring an in-depth analysis of available vehicle sensor data, including temperature, RPM, and pressure readings. The system architecture was designed to ensure seamless integration between the machine learning model, backend API, and frontend interface. The UI/UX design phase focused on creating an intuitive and user-friendly interface to display real-time diagnostics and maintenance predictions effectively. Various wireframes and mockups were developed and iterated based on feedback from potential users.

Additionally, the selection of suitable communication protocols for real-time data transfer was a significant consideration. The project explored the use of MQTT and WebSockets for efficient data streaming. Security measures were also planned to ensure encrypted data transmission and prevent unauthorized access to vehicle diagnostics. Moreover, scalability was taken into account, allowing future expansion of the system to handle multiple vehicle types and integrate additional predictive models.

5.2 Implementation:

The implementation phase involved setting up the infrastructure for data collection, model training, and frontend-backend integration. Initially, the data pipeline was constructed to collect sensor readings in real time and store them in a structured database. Data preprocessing steps such as cleaning, normalization, and feature extraction were performed before feeding the data into the GBM model for training.

The GBM model was trained on historical maintenance data to ensure accurate predictions. Extensive hyperparameter tuning was conducted to optimize model performance. The backend, built using Flask, acted as an API to serve model predictions to the frontend. The React-based web application was designed to provide an interactive dashboard where users could visualize real-time sensor data, monitor vehicle health status, and receive predictive maintenance alerts.

Deployment strategies were also considered, ensuring the system could operate efficiently in cloud and on-premise environments. A continuous monitoring system was set up to track model performance, detect data drift, and trigger automatic retraining when necessary. Finally, rigorous testing was performed, including unit testing, integration testing, and user acceptance testing, to validate system reliability and accuracy.

The GBM model was trained on historical maintenance data, and the React-based web interface was developed for real-time visualization. Flask was used as a backend API to serve predictions.

6. Tools and Techniques Used

6.1 Tools:

- **React.js:** Frontend framework used to build an interactive user interface for real-time data visualization.
- **Spring Boot:** Used for backend development, handling API requests, database management, and system integration.
- **TabNet Model:** Deep learning model used for predictive maintenance and fault detection in vehicles.
- **LLaMA (LLM):** A large language model utilized to analyze sensor readings and generate diagnostic insights.

6.2 Techniques:

- **Feature Engineering:** Extracting meaningful insights from raw sensor data, normalizing values, and identifying key indicators for failure prediction.
- **Hyperparameter Tuning:** Optimizing TabNet model performance using grid search and Bayesian optimization.
- **Time-Series Analysis:** Used for trend detection in vehicle health monitoring and breakdown prediction.
- **Deep Learning-based Fault Detection:** Leveraging TabNet's sequential attention mechanism for feature selection and anomaly detection.
- **Natural Language Processing (NLP):** Using LLaMA to interpret sensor readings and provide actionable recommendations.

7. Results and Discussion

7.1 Final Results:

- The TabNet model successfully predicted vehicle breakdowns and maintenance timelines with high accuracy.
- The LLM component provided valuable insights on detected issues and recommended precautionary measures.
- The real-time visualization dashboard displayed sensor trends, fault predictions, and recommended actions.
- Testing demonstrated reduced downtime and improved maintenance scheduling efficiency.

7.2 Discussion:

- The objectives of the project were met, with accurate predictions and effective recommendations.
- The integration of LLM allowed for improved interpretability of sensor data and better user decision-making.
- Some unexpected outcomes included sensor noise affecting prediction accuracy, which required advanced filtering techniques.
- Future improvements include refining the model with more diverse vehicle datasets and improving system latency.

8. Prototype (Hardware/Software)

8.1 Prototype Description:

- The system consists of a Spring Boot backend, a React.js frontend, and a TabNet-powered AI model integrated with an LLM for insights.
- The web dashboard provides real-time monitoring, fault detection, and maintenance scheduling suggestions.

8.2 Development Process:

- The backend and frontend were developed in parallel to streamline API integration.
- Model training involved collecting and preprocessing real-time vehicle sensor data.
- Challenges faced included data inconsistencies and sensor connectivity issues, which were resolved through rigorous preprocessing and improved hardware configurations.

8.3 Testing and Validation:

- Unit and integration testing were conducted to verify the functionality of each system component.
- Model performance was evaluated using accuracy, precision, recall, and F1-score metrics.
- The system was tested on multiple vehicles to assess its generalization capability, with promising results.
- User feedback was incorporated to improve the UI/UX and make recommendations more interpretable.

9. Conclusion

9.1 Summary:

- The project successfully developed an AI-powered predictive maintenance system for real-time vehicle diagnostics.
- Key objectives, including fault detection, predictive maintenance, and real-time data visualization, were achieved.
- The integration of TabNet and LLaMA significantly enhanced model interpretability and prediction accuracy.

9.2 Personal Reflection:

- The project enhanced our understanding of deep learning, NLP, IoT integration, and real-time data processing.
- Each team member gained hands-on experience in software development, AI model deployment, and cloud-based deployment strategies.
- Future work involves refining model accuracy, extending hardware support, and incorporating additional sensor inputs.

10. Visuals:

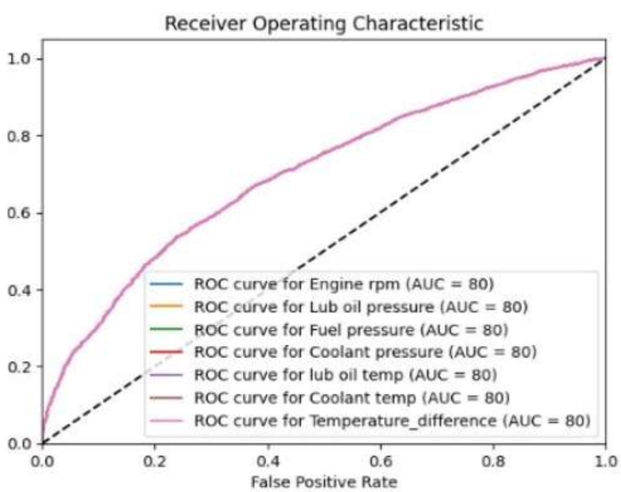

Fig 4: ROC Curve

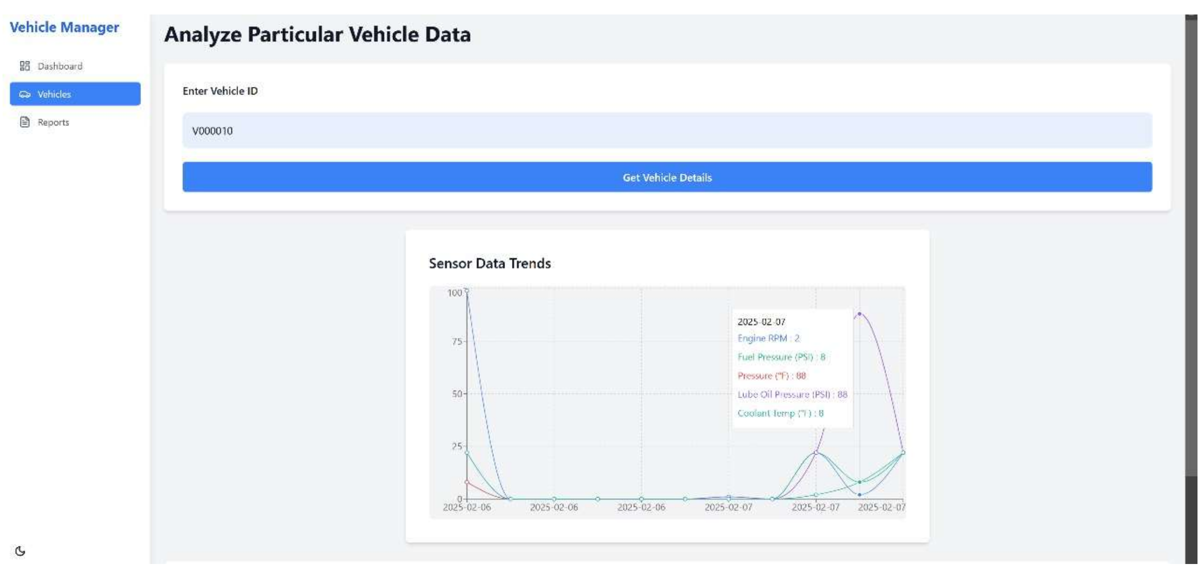

Fig 5: Final Results

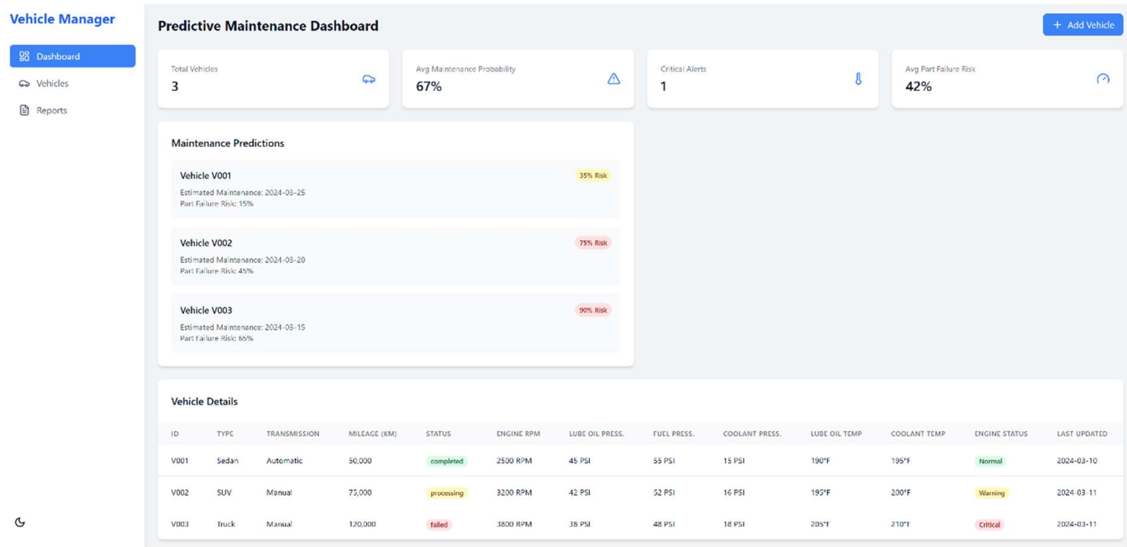

Fig 6: Final result

11. QR Code of Demonstration Video