BE PROJECT SEMESTER VII REPORT

for

# AUTOMOTIVE DIAGNOSTIC AND ACTIVITY MONITORING SYSTEM


| | |
|---|---|
| Chetan Naik | 21CE17 |
| Gaurang Madkaikar | 21CE20 |
| Mangesh Phadte | 21CE36 |
| Travis Samson Fernandes | 21CE72 |


BACHELOR OF ENGINEERING

IN

COMPUTER ENGINEERING

OF GOA UNIVERSITY

**2024 - 2025**


COMPUTER ENGINEERING

PADRE CONCEICAO COLLEGE OF ENGINEERING

VERNA GOA – 403722

**PADRE CONCEICAO COLLEGE OF ENGINEERING**

**VERNA GOA – 403722**

**BE PROJECT SEMESTER VII REPORT**

**for**

# AUTOMOTIVE DIAGNOSTIC AND ACTIVITY MONITORING (ADAM) SYSTEM



| | |
|---|---|
| **Chetan Naik** | **21CE17** |
| **Gaurang Madkaikar** | **21CE20** |
| **Mangesh Phadte** | **21CE36** |
| **Travis Samson Fernandes** | **21CE72** |



Submitted as a requirement for Semester VII Project examination.

**2024 - 2025**

**Approved By:**

..............................................

**Vijaykumar Naik Pawar**
Guide

..............................................  ..............................................

**Dr. Supriya Patil** — Head of the Department

**Dr. Mahesh Parappagoudar** — Principal

**Examined By:**

**Examiner 1:** ..........................................................................................

**(Name / Signature / Date)**

**Examiner 2:** ..........................................................................................

**(Name / Signature / Date)**

# CONTENTS

# ACKNOWLEDGMENT

We take this opportunity to whole-heartedly thank those who were instrumental in helping us achieve our goals. We extend our heartfelt gratitude to our Principal, **Dr. Mahesh Parappagoudar**, for his unwavering support and encouragement throughout the duration of this project. His visionary leadership and constant motivation have been invaluable to us.

We are equally grateful to our Head of Department, **Dr. Supriya Patil**, for providing us with the necessary resources, facilities, and guidance that played a crucial role in the successful completion of our work. We would like to express our deepest appreciation to our project guide, **Vijaykumar Naik Pawar**, whose insightful suggestions, constructive feedback, and dedicated support were instrumental in shaping this project. Their expertise and constant encouragement inspired us to push our limits and achieve the desired outcomes.

We would also like to thank the staff of the **Department of Computer Engineering** for their cooperation and support. We also wish to acknowledge the valuable contributions of the faculty members, whose knowledge and guidance have enriched our understanding throughout the course of this endeavour. Additionally, we are thankful to our peers and friends for their encouragement, collaboration, and constructive feedback, which motivated us to refine our work and strive for excellence.

# ABSTRACT

The Automotive Diagnostic and Activity Monitoring (ADAM) System is a ground-breaking initiative aimed at revolutionizing the way vehicles are operated, monitored, and maintained. Leveraging advanced technologies, ADAM seeks to enhance road safety, streamline user convenience, and offer robust forensic support. By collecting and analysing critical vehicular data prior to accidents, the system proactively identifies and addresses potential risks, reducing the likelihood of road incidents. This preventive mechanism not only safeguards drivers and passengers but also contributes to building safer roadways through timely and effective risk mitigation strategies.

One of the standout features of the ADAM System lies in its utility for forensic investigations. By automating the monitoring and recording of vehicle activities, the system becomes an indispensable tool for law enforcement agencies and insurance companies. It facilitates the reconstruction of accident scenarios with high accuracy, aiding in the determination of liability and the resolution of disputes. This functionality ensures greater transparency and fairness in investigations, accelerating claims processing and fostering trust among stakeholders.

In addition to its safety and forensic applications, the ADAM System empowers vehicle owners by providing advanced diagnostic tools to detect and address mechanical issues with precision. By equipping users with reliable insights into their vehicle's condition, it enables informed maintenance decisions, protects them from potential scams by unscrupulous mechanics, and minimizes repair costs. These capabilities ensure that vehicles remain in peak operating condition, reducing downtime and extending their lifespan.

The ADAM System represents a paradigm shift in vehicle operation by seamlessly integrating data collection, real-time analysis, and system automation. Its intelligent features not only enhance vehicle security but also simplify routine processes, making it an indispensable tool for modern-day vehicle owners. By prioritizing safety, convenience, and efficiency, ADAM aspires to redefine the relationship between drivers and their vehicles. Through its innovative and user-focused approach, the system aims to create a driving experience that is safer, smarter, and more intuitive, ultimately contributing to a more connected and secure transportation ecosystem.

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| Name | Description |
|---|---|
| ADAM: | Automotive Diagnostic and Activity Monitoring |
| OBD: | On-Board Diagnostics Port |
| DTC: | Diagnostic Trouble Codes |
| ECU: | Engine Control Unit |
| RTO: | Regional Transport Office |
| GPS: | Global Positioning System |
| IAT: | Intake Air Temperature |
| MAF: | Mass Air Flow |

# INTRODUCTION

The Automotive Diagnostic and Activity Monitoring (ADAM) System is a pioneering project designed to enhance vehicle safety, improve maintenance efficiency, and support forensic investigations. By leveraging technologies such as Raspberry Pi, machine learning, and mobile app integration, ADAM provides real-time diagnostics, driving insights, and data collection capabilities. This comprehensive system aims to empower users, promote safer roads, and establish transparency in automotive operations.

## 1.1 PURPOSE

The Automotive Diagnostic and Activity Monitoring (ADAM) System is designed to redefine vehicle safety, user convenience, and forensic capabilities through innovative technology. By providing real-time diagnostics and driving insights, ADAM aims to promote safer roads, reduce risks, and ensure efficient vehicle maintenance. The system is also equipped to collect and analyze forensic data for post-accident investigations, aiding in uncovering critical information for law enforcement, insurance companies, and vehicle owners.

ADAM empowers users with the ability to understand and monitor their vehicles, enabling informed decisions about maintenance and repairs while safeguarding against fraudulent practices in the automotive repair industry. This dual focus on proactive prevention and forensic analysis establishes the ADAM System as a transformative tool in modern automotive management.

## 1.2 PROJECT SCOPE

The ADAM System leverages a blend of hardware and software technologies, including Raspberry Pi, machine learning algorithms, and mobile app integration, to deliver comprehensive solutions for vehicle monitoring and diagnostics. The scope of the project encompasses:

### 1.2.1 Real-Time Engine Diagnostics and Alerts

The system continuously retrieves critical data from the vehicle's OBD-II port, such as engine performance metrics, error codes, and fuel efficiency. Real-time analysis of this data ensures that users are promptly alerted to potential issues, enabling timely interventions to prevent severe damage or unsafe conditions.

### 1.2.2 Driving Pattern Analysis for Insurance and Government Use

By analyzing driving behaviors such as speed, acceleration, and braking patterns, ADAM provides valuable insights for insurance companies to customize premiums based on driver performance. Additionally, these analytics can support government initiatives in traffic safety and urban planning.

### 1.2.3 Enhanced Vehicle Safety and Accident Analysis

Through advanced data collection and secure storage, ADAM functions as a black box for vehicles. It captures pre-accident data, which can be invaluable for forensic investigations, helping to establish circumstances leading to an incident and enhancing overall road safety.

By integrating cutting-edge technologies, the ADAM System aims to revolutionize how vehicles are operated, maintained, and monitored, ensuring a safer and more transparent experience for all stakeholders.

## 1.3 LITERATURE SURVEY

### 1.3.1 OVERVIEW

The development of the ADAM System draws on extensive research into OBD-II standards, vehicle diagnostics, and data security. The goal was to identify methodologies and technologies that ensure precise data collection, effective analysis, and secure handling. Studies emphasize the role of standardized protocols like PIDs and CAN in creating robust systems for automotive diagnostics and monitoring.

A thorough review of relevant research papers has refined the project's approach, combining proven methodologies with innovative applications. Key areas include vehicle data collection techniques, the integration of embedded systems like Raspberry Pi, and applications of machine learning for real-time diagnostics and forensic analysis.

### 1.3.2 Key Research Insights

1. **OBD-II Protocols and Standards**: Le Nguyen et al. (2024) highlight the importance of using standardized Parameter Identifications (PIDs) and CAN protocols for accurate data retrieval and diagnostics. Their research demonstrated the integration of e-black boxes with GPS and cameras to collect critical data for accident investigations, vehicle performance analysis, and driving behavior monitoring.

2. **Embedded System Integration**: Sawant & Mane (2024) explored the use of MSCAN modules with KEAZ128 microcontrollers for vehicle diagnostics. Their system focused on detecting and monitoring malfunctions using wireless communication between the OBD-II connector and external systems. The study's experiments underscored the reliability of integrating microcontrollers and suggested future expansions to include additional diagnostic capabilities.

3. **Data Security and Analysis**: Multiple studies underscore the importance of securing vehicle data using encryption techniques during storage and transmission. This ensures the integrity of sensitive information and prevents unauthorized access, which is vital for both user trust and forensic validity.

### 1.3.3 Applications of Research

The literature survey directly informed the ADAM System's design andimplementation. By leveraging standardized diagnostic protocols, integrating robustembedded systems, and adopting advanced data encryption, the ADAM System ensuresaccurate, reliable, and secure diagnostics. These insights have guided the creation of aversatile solution for real-time vehicle monitoring and forensic support.

Table 1.1: Literature Review Summary

| **PAPER TITLE** | **DESCRIPTION** | **EXPERIMENT/ RESULTS** | **FUTURE WORK** |
|---|---|---|---|
| The Design and Implementation of New Vehicle Black Box Using The OBD Information<br><br>Duy Le Nguyen, Myung-Eui Lee, Artem Lensky<br><br>(IEEE Xplore) | The Paper discusses E-Black Box that is a combination of a black box and an event-driven recorder that communicates with a vehicle's ECU module through the OBD-II port to collect vehicle status information. It gathers data from various sources, including the OBD-II port, a camera, GPS, and a 9-degree-of-freedom inertial measurement unit (IMU).<br><br>This data is logged in real-time to a Secure Digital (SD) card at intervals of 100 milliseconds. The recording process stops when the vehicle stops or an accident occurs and if the memory becomes full, the oldest data is automatically overwritten to ensure continuous operation.<br><br>The E-Black Box comes with two simulation modes: an online simulation mode that runs in real-time when the vehicle is in operation and an offline simulation mode for debugging and analysis of recorded data. It also comes with smartphone connectivity via Bluetooth, which allows it to send the vehicle's status and location in case of an accident. It can also search for nearby service centers or hospitals based on its status and location. | Accident Investigation, Vehicle Performance, Driver Behavior Analysis | Road tracking and obstacle detection algorithms |

| | | | |
|---|---|---|---|
| Design and Development of On-Board Diagnostic (OBD) Device for Cars<br><br>Pooja Rajendra Sawant, Yashwant B Mane<br><br>(IEEE Xplore) | This research paper discusses the design and development of an On-Board Diagnostic (OBD) system for cars, based on OBD-II standards established by the Society of Automotive Engineers (SAE). OBD-II standards have been mandatory for all cars sold after 1996. The system provides real-time information about a vehicle, such as engine speed, coolant temperature, pressure, and Diagnostic Trouble Codes (DTCs).<br><br>DTCs are five-character codes that have a predefined structure, helping users know the actual status of their vehicles in real time and diagnose malfunctions. Codes can be generic or manufacturer-specific. OBD systems communicate with the CAN bus to enable the vehicle's ECU to communicate with the OBD device through the ISO 15765 standard.<br><br>The device, according to the paper, employs a KEAZ128 microcontroller, an automotive-grade controller, to interpret the OBD-II protocol and act as a bridge between the vehicle and the user's computer. | Wireless communication between OBD connector and PC using Bluetooth, user-friendly GUI | Adding more diagnostic features and data into the device database |

| | | | |
|---|---|---|---|
| Design and Implementation of a Wireless OBD II Fleet Management System<br><br>Reza Malekian, Ntefeng Ruth Moloisane, Lakshmi Nair, BT(Sunil) Maharaj, Uche A.K. Chude-Okonkwo<br><br>(IEEE Xplore) | This paper introduces a wireless system for fleet management, monitoring vehicle speed and fuel consumption, and computing distance by using an OBD-II module along with GPS data. The system uses the ELM327 IC for decoding OBD-II signaling protocols and communicates using AT commands. The OBD-II protocols implemented are ISO 15765 (CAN), ISO 9141-2, and ISO 14230-4.<br><br>In this system, speed is coded as "0D" in hexadecimal, and MAF is "10." The fuel is calculated based on the measured MAF, and distance is calculated based on the measured speed. This information is logged along with the GPS data simultaneously.<br><br>All logged data is transmitted to a remote server via a Carambola2 WiFi module, where it is stored in an SQL database. The analysis of the data is made through a graphical interface. The system was tested on a BMW 125i and an OBD-II emulator. The measurements of distance had an error margin of about 5%, mainly due to a 1-second data sampling interval. | Tested with OBD II emulator and a real vehicle. Able to accurately measure speed, fuel consumption, and distance with approximately 5% error over short and long distances. Communication range is 900 meters for WiFi. | Proposed using GSM instead of WiFi for wider range, adding a backup battery to maintain wireless communication when the vehicle is off, and improving initialization time. |

| | | | |
|---|---|---|---|
| Machine Learning Based Automatic Diagnosis in Mobile Communication Networks<br><br>Kuo-Ming Chen, Tsung-Hui Chang, Kai-Cheng Wang, Ta-Sung Lee<br><br>(IEEE Xplore) | This paper reports on an ML-based algorithm for automatic fault diagnosis in mobile communication networks. The scheme utilizes SONs, removing the need for expensive hardware by making the networks self-configurable, self-organizing, and self-healing. The presented methodology combines a Softmax NN and SVM models with the aim of improving diagnosis accuracy.<br><br>It further uses KPIs as well as performance management counters (PMCs) that are used in feature extraction. The combined output from the Softmax NN along with SVM carries out multiclass classification while diagnosing network faults accordingly. Thus the ML-based algorithms are of unsupervised type because they learn the training datasets directly and thereby are malleable in real application scenarios.<br><br>This dual capability addresses both single and multi-fault scenarios, making the system robust enough for modern mobile networks as an efficient alternative to fault-detection methods. | Achieves high accuracy in both single and multi-fault detection in LTE networks | Explore other models and enhance multi-fault detection. |

| | | | |
|---|---|---|---|
| Vehicular Mobile Commerce<br><br>Authors [Not provided in excerpt]<br><br>(IEEE Xplore) | Wi-Fi-enabled vehicles enables m-commerce applications, like entertainment and diagnostics. Significant technical, safety, and financial challenges remain | Highlights real-world prototypes, such as Ford's Lincoln and an Atlanta traffic data project. These illustrate potential but underscore connectivity and safety issues. | Calls for solutions to improve stable computing tech in high-speed connectivity, security, privacy, and cost efficiency to make vehicular m-commerce practical and safe. |

| | | | |
|---|---|---|---|
| Automotive Internal-Combustion-Engine Fault Detection and Classification Using Artificial Neural Network Techniques<br><br>Ryan Ahmed, Mohammed El Sayed, S. Andrew Gadsden, Jimi Tjong, Saeid Habibi<br><br>(IEEE Xplore) | This research examines the use of artificial neural networks for the detection and classification of faults in internal combustion engines. The paper has trained such ANNs based on the vibration data measurements in the crank angle domain and with the potential to set up a consistent diagnostic system for dealerships and assembly plants, reducing the cost of manufacturers' warranty.<br><br>Methodologies in the paper involve feedforward multilayer perceptron ANNs, with the appeal for these learning, adapting, noise rejecting, and handling nonlinear relations. Training algorithms used here include backpropagation, Levenberg-Marquardt, quasi-Newton, extended Kalman filter, and the newest variable structure filter introduced as a smooth variable structure filter. The variable structure filter is notable because of stability, robustness, and the good response to the uncertainties. | SVSF-ANN shows high accuracy and reliability for fault classification. | Expand to real-time applications and add more engine parameters. |

| | | | |
|---|---|---|---|
| Automotive Internal-Combustion-Engine Fault Detection and Classification Using Artificial Neural Network Techniques | The research benefits the detection process of faults by avoiding special location models of fault. The test bench employed a semi-anechoic chamber with a four-stroke eight-cylinder engine. Triaxial accelerometers were used to collect data of vibrations and then transform this data into the crank angle domain by the help of a cylinder identification sensor. The tested faults included missing bearings, piston chirps, chain tensioners, etc., each of which possesses some specific vibration patterns. | SVSF-ANN shows high accuracy and reliability for fault classification. | Expand to real-time applications and add more engine parameters. |
| Deep Learning Model Based CO2 Emissions Prediction Using Vehicle Telematics Sensors Data<br><br>Authors [Not provided in excerpt]<br><br>(IEEE Xplore) | This paper presents an LSTM model that predicts CO2 emissions from OBD-II sensor data in real-time. The model is designed to handle noisy data and improve time-series prediction accuracy. The study demonstrates the model's effectiveness in predicting emissions and suggests its potential for broader deployment in real-time monitoring systems. | Outperforms other methods in handling noisy data and time-series prediction accuracy. | Use diverse datasets for broader deployment in real-time monitoring systems. |

| | | | |
|---|---|---|---|
| OBD SecureAlert: An Anomaly Detection System for Vehicles<br><br>Sandeep Nair Narayanan, Sudip Mittal, Anupam Joshi<br><br>(IEEE Xplore) | This paper presents OBD-SecureAlert, a data-driven anomaly detection system for cars designed to identify abnormal behaviors using CAN bus data. The CAN bus is an internal communication network found in vehicles that connects ECUs, sensors, and actuators. Since most CAN buses lack built-in security measures, they are open to cyberattacks that can compromise the safety of automobiles.<br><br>OBD-SecureAlert eliminates this vulnerability by utilizing a Hidden Markov Model (HMM) for anomaly monitoring and detection against deviations from normal vehicle behavior. This system emphasizes detection instead of prevention, thereby creating an applicative solution to identify possible threats. It collects data from the CAN bus, which carries the values of speed and RPM sensors in the vehicle. After being trained, it analyzes the incoming data using a sliding window and flags data sets based on predefined threshold limits as anomalies or safety/cybersecurity threats. They simulate attack scenarios, like injecting anomalous data into the CAN bus, to detect anomalies. | Effectively uses a Hidden Markov Model (HMM) to detect anomalies in vehicle behavior by analyzing CAN bus data, successfully identifying threats like malicious data injections in real-time. | Integrating with machine learning models like LSTM or Transformer-based architectures for anomaly detection or real-time threat mitigation against early signs of wear or failure. |

| | | | |
|---|---|---|---|
| Assessing the Impact of Driving Behavior on Instantaneous Fuel Consumption<br><br>Javier E. Meseguer, Carlos T. Calafate, Juan Carlos Cano, Pietro Manzoni<br><br>(IEEE Xplore) | This paper talks about DrivingStyles, a platform whose aim is to analyze and improve driving habits to diminish fuel consumption and greenhouse gases. It uses data directly from the vehicle's Electronic Control Unit (ECU), accessed by an OBD-II Bluetooth interface, and smartphone technology to assess driving behavior and the efficiency of fuel.<br><br>The research addresses the rising fuel costs and environmental pollution caused by greenhouse gas emissions. Applying data mining and neural networks, the platform classifies driving styles based on speed, acceleration, and engine RPM. It calculates fuel consumption using data from MAF, MAP, and IAT sensors, with alternative methods for cases where direct data is unavailable.<br><br>The system consists of an Android app, an OBD-II interface, and a web-based data center. The app collects data from the vehicle and phone sensors and sends it to the data center for analysis. A neural network identifies driving styles and road types, while the data center visualizes fuel efficiency and habits using open-source tools.<br><br>The platform also calculates CO2 emissions based on fuel consumption. The study reveals that driving style is significantly affecting the fuel usage and emissions. Aggressive driving is found to consume more fuel, and it can be up to 15–20% saved in fuel by adopting efficient driving practices. | An efficient driving style can drastically reduce fuel consumption and reduce greenhouse gas emissions, with an estimated saving of 15–20%. The DrivingStyles platform, with smartphone and vehicle data, is one of the practical tools to help raise awareness and improve driving styles for better energy efficiency. | Long Short-Term Memory (LSTM) networks or Transformer models to better capture temporal driving patterns and develop individualized driving style profiles based on user-specific data to provide tailored recommendations. |

| | | | |
|---|---|---|---|
| Exploring Fuzzy Logic and Random Forest for Car Drivers' Fuel Consumption Estimation in IoT-Enabled Serious Games<br><br>Rana Massoud, Francesco Bellotti, Riccardo Berta, Alessandro De Gloria, Stefan Poslad<br><br>(IEEE Xplore) | This paper explores the use of the Fuzzy Logic (FL) and Random Forest (RF) algorithms to estimate vehicle fuel consumption in the context of providing foundational support for the development of an IoT-enabled driving game which sends real-time feedback to promote good driving habits and minimize consumption using the enviroCar database.<br><br>This research employs real-time vehicle data that are collected through the OBD-II interface to establish accurate models for FC estimation. Important variables include throttle position sensor (TPS), revolutions per minute (RPM), car speed, and fuel consumption (FC) and tested on a range of car models and driving environments is considered in order to make the system applicable in real-world applications.<br><br>FL uses "if-then" rules to provide understandable feedback, while RF applies machine learning for high-accuracy predictions. Results show RF outperforms FL, achieving higher $R^2$ (0.896 vs. 0.65) and lower MSE (1.506 vs. 4.745). Both models operate in real-time, with RF emphasizing speed and RPM as critical predictors.<br><br>Combining FL for intuitive coaching with RF for precise FC estimates should prove to improve IoT-enabled driving games. This solution should even steer game designers and also scale to other IoT applications which require real-time appraisals. | Fuzzy Logic (FL) provides interpretable feedback for coaching, while Random Forest (RF) delivers superior predictive accuracy for fuel consumption modeling. The combination of both models benefits from their strengths and makes them ideal for IoT-enabled driving games and applicable to other fields requiring real-time user performance assessment. | Benchmark the performance of RF and FL against newer algorithms, such as Gradient Boosting (e.g., XGBoost) or neural networks, to validate the choice of models. |

# OVERALL DESCRIPTION

This chapter delves into the Automotive Diagnostic and Activity Monitoring (ADAM) System, providing an in-depth exploration of its innovative design, core functions, and user-centric features. It outlines the project's integration of advanced hardware and software components to offer comprehensive solutions in vehicle diagnostics, maintenance, and forensic analysis.

## 2.1 PROJECT PERSPECTIVE

The Automotive Diagnostic and Activity Monitoring (ADAM) System is a standalone innovation designed to replace traditional diagnostic tools with advanced, integrated solutions. It communicates directly with the car's Engine Control Unit (ECU) via OBD-II protocols, offering a seamless and efficient interface. By collecting real-time data and facilitating in-depth analysis, ADAM provides vehicle owners, insurers, and government bodies with actionable insights to enhance safety, maintenance, and operational transparency.

Unlike conventional diagnostic tools, the ADAM System combines hardware components, such as a Raspberry Pi and GPS modules, with sophisticated software algorithms for predictive diagnostics and forensic capabilities. This ensures reliability, user convenience, and versatility across various use cases. The system's ability to integrate seamlessly with a variety of vehicle models makes it a scalable and forward-compatible solution.

## 2.2 Project Functions

1. **Black Box Data Collection:** The system records key parameters, including RPM, speed, throttle position, engine temperature, and error codes. This data is stored securely and can be analyzed to understand vehicle performance and pre-accident conditions. By functioning as a black box, ADAM serves as a reliable source of information for investigations, aiding in uncovering critical details about vehicle operations prior to incidents.

2. **Mobile App Visualizations:** Users can access real-time vehicle statistics, diag-

nostics, and alerts through an intuitive mobile application. The app also provides detailed visualizations and suggestions for maintenance, ensuring users are informed and empowered to take necessary actions. Additionally, the app integrates with cloud services to enable historical data access, trend analysis, and remote monitoring.

3. **Forensic Data Sharing:** ADAM facilitates secure export of vehicle data in CSV format for use by insurance companies and law enforcement agencies. This feature supports claim verification and accident investigations, providing transparency and accountability. Advanced encryption ensures that data remains confidential during transmission, preserving the integrity of sensitive information.

## 2.3 User Classes and Characteristics

1. **Vehicle Owners:**
   - Monitor real-time vehicle health and performance metrics.
   - Receive proactive alerts for maintenance and potential issues.
   - Access diagnostic tools to understand and resolve problems effectively.
   - Benefit from enhanced transparency in repair processes, reducing the risk of being misled by mechanics.
2. **Insurance Agents:**
   - Utilize driving data to verify claims and assess driver behavior.
   - Access pre-accident data to establish liability and streamline claim processing.
   - Leverage aggregated data insights for policy adjustments and risk assessment.
3. **Government Bodies:**
   - Analyze anonymized driving habits to inform traffic safety policies.
   - Use aggregated data to identify patterns and improve urban planning.
   - Incorporate ADAM's analytics into broader smart city initiatives to optimize transportation networks.

## 2.4 Operating Environment

1. **Hardware Components:**
   - **Raspberry Pi 5:** Serves as the core processing unit for data collection and analysis, capable of handling high volumes of real-time data efficiently.
   - **OBD-II Reader:** Interfaces with the vehicle's ECU to retrieve critical diagnostic data. It supports multiple protocols for compatibility with diverse vehicle models.
   - **GPS Module:** Tracks vehicle location and assists in route analysis and forensic investigations, providing geographic context to vehicle data.
   - **Storage Devices:** Ensures secure and encrypted storage of vehicle data for analysis and sharing. Options include SD cards and external drives for scalability.
2. **Software Components:**
   - **Raspberry Pi OS:** The operating system managing hardware functions and software execution, optimized for real-time processing tasks.
   - **Python Libraries:** TensorFlow for predictive analytics, Flask for application interfacing, and encryption libraries for secure data handling. Libraries for GPS data processing and CAN protocol decoding are also integrated.
   - **Mobile Application:** Provides an interface for end-users to access vehicle data, visualize statistics, and receive alerts. Features include cloud synchronization, customizable dashboards, and multi-user support for fleet management.

# REQUIREMENTS

## 3.1 Functional Requirements

Functional requirements define the specific operations and functionalities the ADAM System must perform under various conditions. These include:

1. **Real-Time Data Collection:**
    - Continuously retrieve and store data such as RPM, speed, engine temperature, and throttle position from the OBD-II interface.
    - Ensure secure and uninterrupted data transmission to the processing unit.
2. **Black Box Functionality:**
    - Record and retain critical vehicle parameters during regular operation and immediately preceding an accident.
    - Enable playback or export of stored data for analysis and forensic purposes.
3. **Mobile Application Features:**
    - Provide real-time visualizations of vehicle performance and health metrics.
    - Alert users to diagnostic issues and suggest maintenance actions.
    - Support CSV export of forensic data for law enforcement or insurance purposes.
4. **Driving Pattern Analysis:**
    - Collect and analyze driving behavior, including acceleration, braking, and speed trends.
    - Provide feedback for users, insurers, and government agencies to improve safety and policy planning.
5. **Forensic Data Sharing:**
    - Enable encrypted sharing of critical data with third parties, ensuring compliance with privacy standards.

## 3.2 Other Nonfunctional Requirements

Nonfunctional requirements describe the system's expected performance and quality attributes. These include:

1. **Reliability:**
    - Ensure 99.9
    - Implement fail-safe mechanisms for data storage to prevent loss during system failure.
2. **Security:**
    - Encrypt all stored and transmitted data to protect sensitive user and vehicle information.
    - Restrict access to authorized personnel or applications only.
3. **Performance:**
    - Process and analyze real-time data within 1 second of retrieval.
    - Maintain seamless interaction between the mobile app and backend systems with minimal latency.
4. **Maintainability:**
    - Design modular software components to facilitate updates and bug fixes.
    - Provide comprehensive documentation for developers and users.
5. **Scalability:**
    - Support integration with diverse vehicle models and OBD-II protocols.
    - Ensure compatibility with additional sensors and third-party services in future expansions.

## 3.3 Hardware Requirements

The hardware required to develop and deploy the ADAM System includes:

1. **Raspberry Pi 5:** Serves as the primary processing unit for data collection, analysis, and storage.
2. **OBD-II Reader:** Interfaces with the vehicle's ECU to retrieve diagnostic data.

3. **GPS Module:** Tracks vehicle location for route analysis and forensic applications.

4. **Storage Devices:** SD card or external drives for secure, encrypted data storage.

5. **Power Supply:** Reliable and portable power solutions for the Raspberry Pi and peripherals.

6. **Display Monitor:** For debugging and system setup during development.

## 3.4 Software Requirements

The software required to implement the ADAM System includes:

1. **Raspberry Pi OS** for the main processing unit.

2. **Python** for core logic and data processing.TensorFlow for machine learning and predictive analytics.Flask for building APIs and web-based interfaces.Cryptography libraries for secure data handling.GPS and CAN protocol libraries for location and ECU communication.

3. **React Native** React Native for developing a cross-platform mobile application.

4. **SQLite** a lightweight database for local storage during development.

5. **Supabase** for remote data synchronization and backup.

# SYSTEM DESIGN

## 4.1 System Architecture Overview

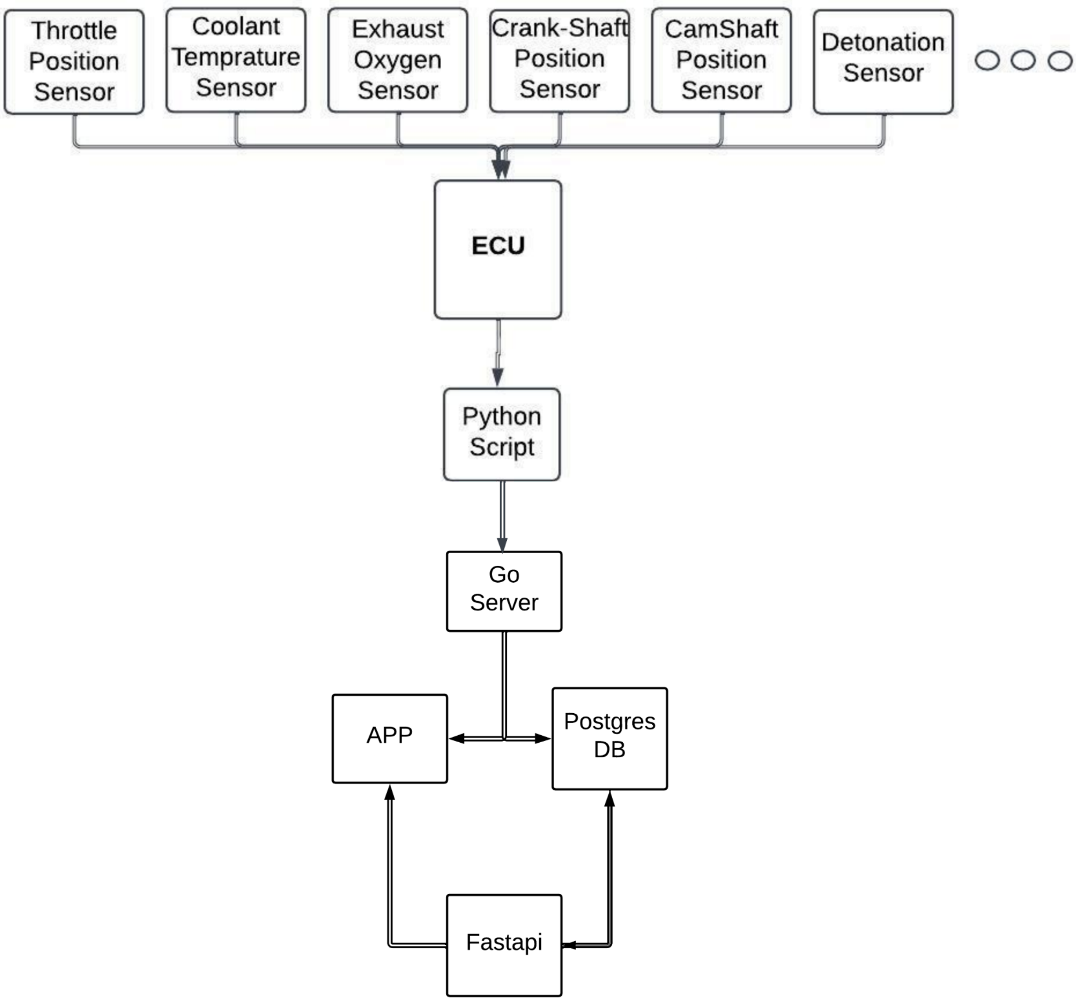


Figure 4.1: System Design Overview

The diagram illustrates the flow of data and interactions within an automotive diagnostic and monitoring system. The system integrates multiple sensors, data processing units, databases, and APIs to achieve real-time diagnostics and data sharing. Below is a detailed explanation of the components and their interactions:

1. **Sensors** The system gathers data from a wide range of vehicle sensors to monitor various aspects of vehicle performance and health:

    - **Throttle Position Sensor:** Monitors the position of the throttle to assess acceleration and deceleration patterns.
    - **Coolant Temperature Sensor:** Measures the temperature of the engine coolant to monitor engine heat levels.
    - **Exhaust Oxygen Sensor (O2 Sensor):** Tracks oxygen levels in the exhaust to evaluate fuel efficiency and combustion quality.
    - **Crankshaft Position Sensor:** Detects the rotational speed and position of the crankshaft to ensure proper engine timing.
    - **Camshaft Position Sensor:** Monitors the camshaft's rotation to synchronize it with the crankshaft.
    - **Detonation Sensor (Knock Sensor):** Identifies engine knock or detonation for optimizing performance and safety.
    - **Mass Air Flow (MAF) Sensor:** Measures the mass of air entering the engine to ensure optimal air-fuel mixture.
    - **Manifold Absolute Pressure (MAP) Sensor:** Tracks pressure in the intake manifold to monitor engine load.
    - **Fuel Pressure Sensor:** Monitors the pressure of fuel in the fuel system to ensure proper delivery to the engine.
    - **Oil Pressure Sensor:** Detects oil pressure levels to prevent engine damage due to insufficient lubrication.
    - **Transmission Fluid Temperature Sensor:** Monitors the temperature of the transmission fluid to ensure efficient operation.
    - **Wheel Speed Sensors:** Measure the rotational speed of each wheel for anti-lock braking and traction control systems.
    - **Brake Pedal Position Sensor:** Tracks the position of the brake pedal for braking system diagnostics.
    - **Steering Angle Sensor:** Monitors the steering wheel's position to assess turning angles and stability control.
    - **Tire Pressure Monitoring System (TPMS) Sensors:** Measure tire pressure to ensure safety and performance.
    - **Ambient Temperature Sensor:** Tracks the temperature outside the vehicle to assist in climate control and engine adjustments.

- **Battery Voltage Sensor:** Monitors the vehicle's battery voltage to detect charging or electrical system issues.

These sensors send real-time data to the Electronic Control Unit (ECU) for initial processing and control actions.

2. **Electronic Control Unit (ECU)** An ECU, or Electronic Control Unit, is the heart and brain of the vehicle, serving as an embedded system in automotive electronics responsible for controlling one or more electrical systems or subsystems in a vehicle. It acts as the central hub that gathers sensor data and error codes, which are accessed via the OBD-II port operating on a client-server architecture. When specific commands are sent to the OBD-II port, it communicates with the ECU and retrieves data in the form of hexadecimal codes. The first two digits of these codes indicate the type of sensor, while the remaining digits represent the corresponding value.
   Modern vehicles often contain numerous ECUs—sometimes over 150—handling various functions such as engine management, transmission control, braking systems, powertrain operations, and more. These ECUs collectively form the vehicle's computer network, enabling diagnostics, advanced functionalities, and seamless control of the vehicle's electrical and mechanical systems.

3. **On-Board Diagnostic Port (OBD Port)** On-Board Diagnostics (OBD) is a vehicle's self-diagnostic and reporting capability that monitors the performance of various vehicle subsystems, primarily focusing on emissions control. This system became a requirement in the United States to comply with federal emissions standards, enabling the detection of failures that could lead to excessive tailpipe emissions. The OBD system provides vehicle owners and technicians with access to diagnostic trouble codes (DTCs), which facilitate the identification of malfunctions within the vehicle.

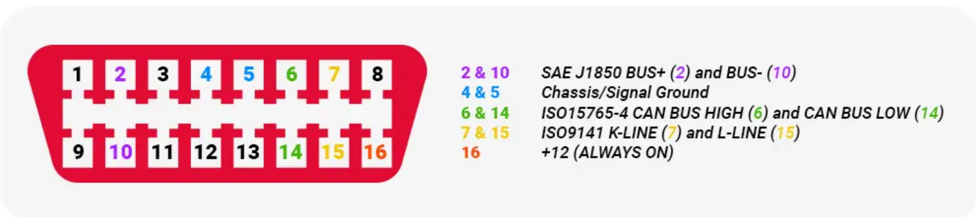


Figure 4.2: OBD-II Connector Pin-out

The OBD-II standard specifies a 16-pin connector known as the J1962 connector, which is used universally across most vehicles. The pinout for this connector is defined as follows:

Table 4.1: OBD-II Connector Pinout

| **Pin** | **Function** |
|---|---|
| 1 | Manufacturer discretion |
| 2 | Bus positive Line (SAE J1850 PWM and VPW) |
| 3 | Manufacturer discretion |
| 4 | Chassis ground |
| 5 | Signal ground |
| 6 | CAN high (ISO 15765-4 and SAE J2284) |
| 7 | K-line (ISO 9141-2 and ISO 14230-4) |
| 8 | Manufacturer discretion |
| 9 | Manufacturer discretion |
| 10 | Bus negative Line (SAE J1850 PWM only) |
| 11 | Manufacturer discretion |
| 12 | Manufacturer discretion |
| 13 | Manufacturer discretion |
| 14 | CAN low (ISO 15765-4 and SAE J2284) |
| 15 | L-line (ISO 9141-2 and ISO 14230-4) |
| 16 | Battery voltage (+12 Volt for type A connector, +24 Volt for type B connector) |

The assignment of unspecified pins is left to the vehicle manufacturer's discretion, allowing for variations in implementation across different brands and models.

The communication between the OBD system and the Engine Control Unit (ECU) occurs through standardized protocols. This communication involves several key components:

Data Link Connector: The OBD-II connector serves as a physical interface for diagnostic tools to communicate with the ECU. Protocols: Various communication protocols are utilized, including:

- ISO 9141-2
- ISO 14230-4 (Keyword Protocol 2000)
- ISO 15765-4 (CAN bus)

Parameter Identification (PID): PIDs are specific data points that can be queried from the ECU. They provide information about various vehicle parameters such as engine RPM, vehicle speed, coolant temperature, etc. Each PID corresponds to a specific function or piece of data that the ECU can report back to diagnostic tools.

4. **Python Script** A Python script retrieves data from the ECU and processes it for further analysis or storage. It acts as a bridge between the raw sensor data and the database/server layer.

5. **Postgres Database** The Postgres database serves as the primary storage for processed telemetry data. It provides a structured and reliable backend for managing sensor data and results from further computations.

6. **Go Server** The Go server processes data stored in the Postgres database and performs additional tasks, such as:

   - Data aggregation
   - Complex computations or analytics
   - API responses for connected systems

7. **FastAPI Script** The FastAPI script interacts with the Postgres database to:

   - Expose APIs for retrieving or upserting data
   - Enable data queries from other applications or systems
   - Provide endpoints for monitoring, diagnostics, and reporting

8. **Mobile App** The app provides the user interface for interacting with the system. Through the integration with a Golang server and PostgreSQL database, the app can:

    - Display real-time telemetry data
    - Provide diagnostic reports
    - Enable user interaction and configuration of the system

# IMPLEMENTATION

## 5.1 Data Collection from Car OBD Port

The **Automotive Diagnostic and Activity Monitoring (ADAM)** system utilizes the car's **OBD2 (On-Board Diagnostics)** port to collect a wide range of real-time vehicle data, including critical parameters such as engine RPM, vehicle speed, fuel consumption, and engine load. The type and quantity of data available through the OBD2 interface can vary significantly depending on the car's make, model, and the sensors it supports.

### Sensor Variability and Data Challenges

Each car is equipped with a unique set of sensors, leading to differences in the **Parameter IDs (PIDs)** that are supported. For instance, while one vehicle might provide detailed oxygen sensor readings, another might lack these advanced diagnostics. The ADAM system adapts to this variability by querying the Supported PIDs (PIDS_A, PIDS_B, etc.) to identify which parameters can be accessed for a given vehicle. This ensures effective operation across a broad range of car models.

### Real-Time Data Collection

Although the system collects data in **real-time**, there is a slight delay in transmission and processing. Data points are typically retrieved at intervals of **30 minutes** due to system and hardware limitations, balancing resource efficiency with the need for timely diagnostics. Despite this, the system continuously monitors vehicle performance and updates its diagnostics.

### PIDs And Their Metrics

The following is a subset of the PIDs that the system may query, along with their descriptions and response values:

Table 5.1: Example PIDs and Their Metrics

| PID | Name | Description | Response Value |
|---|---|---|---|
| 1 | STATUS | Status since DTCs cleared | Special |
| 4 | ENGINE_LOAD | Calculated engine load | Unit: Percent |
| 0C | RPM | Engine RPM | Unit: RPM |
| 0D | SPEED | Vehicle speed | Unit: KPH |
| 10 | MAF | Airflow rate (Mass Air Flow) | Unit: grams/second |
| 2C | COMMANDED_EGR | Commanded Exhaust Gas Recirculation | Unit: Percent |
| 5C | OIL_TEMP | Engine oil temperature | Unit: Celsius |
| 59 | FUEL_RAIL_PRESSURE_ABS | Absolute fuel rail pressure | Unit: Kilopascal |

### 5.1.1 Adaptability of the ADAM System

The variability in available PIDs necessitates a flexible system design. The ADAM system is designed to dynamically query the supported PIDs of each vehicle and adjust its data collection and processing pipeline accordingly. This ensures that it can handle everything from basic diagnostics to advanced performance monitoring based on the car's capabilities.

### 5.1.2 Addressing Limitations

While data is collected in delayed intervals, the insights generated by the system are still valuable for **diagnostic analysis** and **predictive maintenance**. Future iterations aim to minimize delays and increase data retrieval speed for a truly real-time experience. By leveraging rich datasets from the OBD2 port and accommodating sensor variability, the ADAM system provides robust and customizable diagnostics for various vehicles.

## 5.2 Data Transfer to Go Server

After data is collected from the car's OBD port, it is sent to a **Go server**. This server acts as middleware to handle data flow efficiently from the Python script to backend systems, ensuring seamless data exchange for real-time monitoring and analysis.

## 5.3 Data Storage in PostgreSQL Database

The collected data is stored in a **PostgreSQL (PGSQL)** database for future processing and analysis. This database serves as a central repository for storing vehicle data, including various parameters like engine temperature, fuel consumption, speed, etc. The structured format allows for efficient querying and data retrieval.

## 5.4 Data Analysis Using FastAPI and Machine Learning

Once stored in PostgreSQL, **FastAPI** is used to access and process this data for various analyses. Machine learning models are applied to perform several tasks:

1. **Predictive Maintenance**: ML algorithms predict potential failures of car components by analyzing historical and real-time data.
2. **Driver Behavior**: The system tracks aspects of driver behavior such as acceleration and braking patterns.
3. **Emission Compliance**: It monitors emission-related parameters to ensure compliance with environmental regulations.
4. **Engine Health Prediction**: Real-time data analysis predicts potential issues like overheating or mechanical failures.

## 5.5 Chatbot for User Interaction

An integrated chatbot allows users to interact with the system regarding their car's performance metrics. It can answer queries such as:

- **Current vehicle health** (e.g., engine status, fuel efficiency)
- **Upcoming maintenance tasks**
- **Emission compliance**

- **Engine health status and alerts**

This feature provides users with a conversational interface for obtaining insights without complex navigation through menus.

## 5.6 Data Visualization via Mobile App

The ADAM system presents vehicle diagnostics through a mobile app interface that visualizes key performance metrics such as:

- **Maintenance schedules and alerts**
- **Driver behavior analytics**
- **Engine health status and predictions**
- **Emission compliance status**

This user-friendly app enables car owners to track their vehicle's performance, receive maintenance notifications, and monitor real-time data for optimal vehicle health management.

# CONCLUSION

The Automotive Diagnostic and Activity Monitoring (ADAM) System represents an innovative solution for improving vehicle safety, operational efficiency, and user convenience. The system combines real-time vehicle diagnostics, advanced data analysis, and seamless mobile application integration to provide actionable insights to various stakeholders, including vehicle owners, insurance agents, and regulatory authorities. By employing cutting-edge technologies such as machine learning algorithms, Raspberry Pi hardware, and the On-Board Diagnostics II (OBD-II) protocol, the ADAM System enables robust vehicle monitoring, fault detection, and forensic analysis capabilities.

The real-time diagnostic functionality of the ADAM System monitors vehicle performance and identifies potential mechanical issues before they escalate into significant problems, thereby reducing maintenance costs and enhancing road safety. The incorporation of machine learning facilitates predictive analytics, enabling the system to offer recommendations for preventive maintenance and optimize vehicle performance. Using Raspberry Pi as a cost-effective and scalable hardware platform allows for efficient data processing and ensures the adaptability of the system to a wide range of vehicles.

Integration with the OBD-II protocol ensures compatibility with modern vehicles and enables the collection of detailed diagnostic data, including engine performance, fuel efficiency, and emissions levels. The ADAM System also leverages a user-friendly mobile application to provide real-time notifications, comprehensive diagnostic reports, and driving insights, making it accessible and practical for everyday users.

Future advancements could expand the system's capabilities by incorporating features such as obstacle detection, which would enhance situational awareness for drivers. Predictive maintenance could be further refined to include more sophisticated failure predictions based on historical and real-time data. Additionally, integrating the system with advanced driver-assistance systems (ADAS) could contribute to autonomous driving technologies by providing critical diagnostic and environmental data. These developments would position the ADAM System as a pivotal technology in the evolution of intelligent and connected vehicles.

# BIBLIOGRAPHY


1. Duy Le Nguyen, Myung-Eui Lee and A. Lensky, **"The design and implementation of new Vehicle Black Box using the OBD information,"** *2012 7th International Conference on Computing and Convergence Technology (ICCCT)*, Seoul, Korea (South), 2012, pp. 1281-1284.

2. P. R. Sawant and Y. B. Mane, **"Design and Development of On-Board Diagnostic (OBD) Device for Cars,"** *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Pune, India, 2018, pp. 1-4, `https://doi.org/10.1109/ICCUBEA.2018.8697833`.

3. R. Malekian, N. R. Moloisane, L. Nair, B. T. Maharaj and U. A. K. Chude-Okonkwo, **"Design and Implementation of a Wireless OBD II Fleet Management System,"** *IEEE Sensors Journal*, vol. 17, no. 4, pp. 1154-1164, Feb. 15, 2017, `https://doi.org/10.1109/JSEN.2016.2631542`.

4. Landry Frank Ineza Havugimana, Bolan Liu, Fanshuo Liu, Junwei Zhang, Ben Li and Peng Wan, **"Review of Artificial Intelligent Algorithms for Engine Performance, Control, and Diagnosis,"** *Energies*, vol. 16, 2023, pp. 1206, `https://doi.org/10.3390/en16031206`.

5. R. Ahmed, M. El Sayed, S. A. Gadsden, J. Tjong and S. Habibi, **"Automotive Internal-Combustion-Engine Fault Detection and Classification Using Artificial Neural Network Techniques,"** *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 21-33, Jan. 2015, `https://doi.org/10.1109/TVT.2014.2317736`.

6. J. B. Porch, C. Heng Foh, H. Farooq and A. Imran, **"Machine Learning Approach for Automatic Fault Detection and Diagnosis in Cellular Networks,"** *2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, Odessa, Ukraine, 2020, pp. 1-5, `https://doi.org/10.1109/BlackSeaCom48709.2020.9234962`.

7. S. N. Narayanan, S. Mittal and A. Joshi, **"OBD_SecureAlert: An Anomaly Detection System for Vehicles,"** *2016 IEEE International Conference on Smart

*Computing (SMARTCOMP)*, St. Louis, MO, USA, 2016, pp. 1-6, https://doi.org/10.1109/SMARTCOMP.2016.7501710.

8. J. E. Meseguer, C. T. Calafate, J. C. Cano and P. Manzoni, **"Assessing the impact of driving behavior on instantaneous fuel consumption,"** *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, USA, 2015, pp. 443-448, https://doi.org/10.1109/CCNC.2015.7158016.

9. R. Massoud, F. Bellotti, R. Berta, A. De Gloria and S. Poslad, **"Exploring Fuzzy Logic and Random Forest for Car Drivers' Fuel Consumption Estimation in IoT-Enabled Serious Games,"** *2019 IEEE 14th International Symposium on Autonomous Decentralized System (ISADS)*, Utrecht, Netherlands, 2019, pp. 1-7, https://doi.org/10.1109/ISADS45777.2019.9155706.

10. T. Liu, G. Yang and D. Shi, **"Construction of Driving Behavior Scoring Model based on OBD Terminal Data Analysis,"** *2020 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT)*, Shenyang, China, 2020, pp. 24-27, https://doi.org/10.1109/ISCTT51595.2020.00012.