VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI -590 018

Project Report on

"Predictive Maintenance for Automobiles using IOT And Machine Learning''

A Project submitted in the partial fulfillment for the award of the degree

Bachelor of Engineering in Electronics and Communication Engineering

Submitted by:

Laxman M Nayanegali	**1AY21EC060**
Keerthan C K	**1AY22EC402**
Anand S S	**1AY21EC012**
Tarun K H	**1AY22EC410**

Under the Guidance of

Mr. Jagadish M

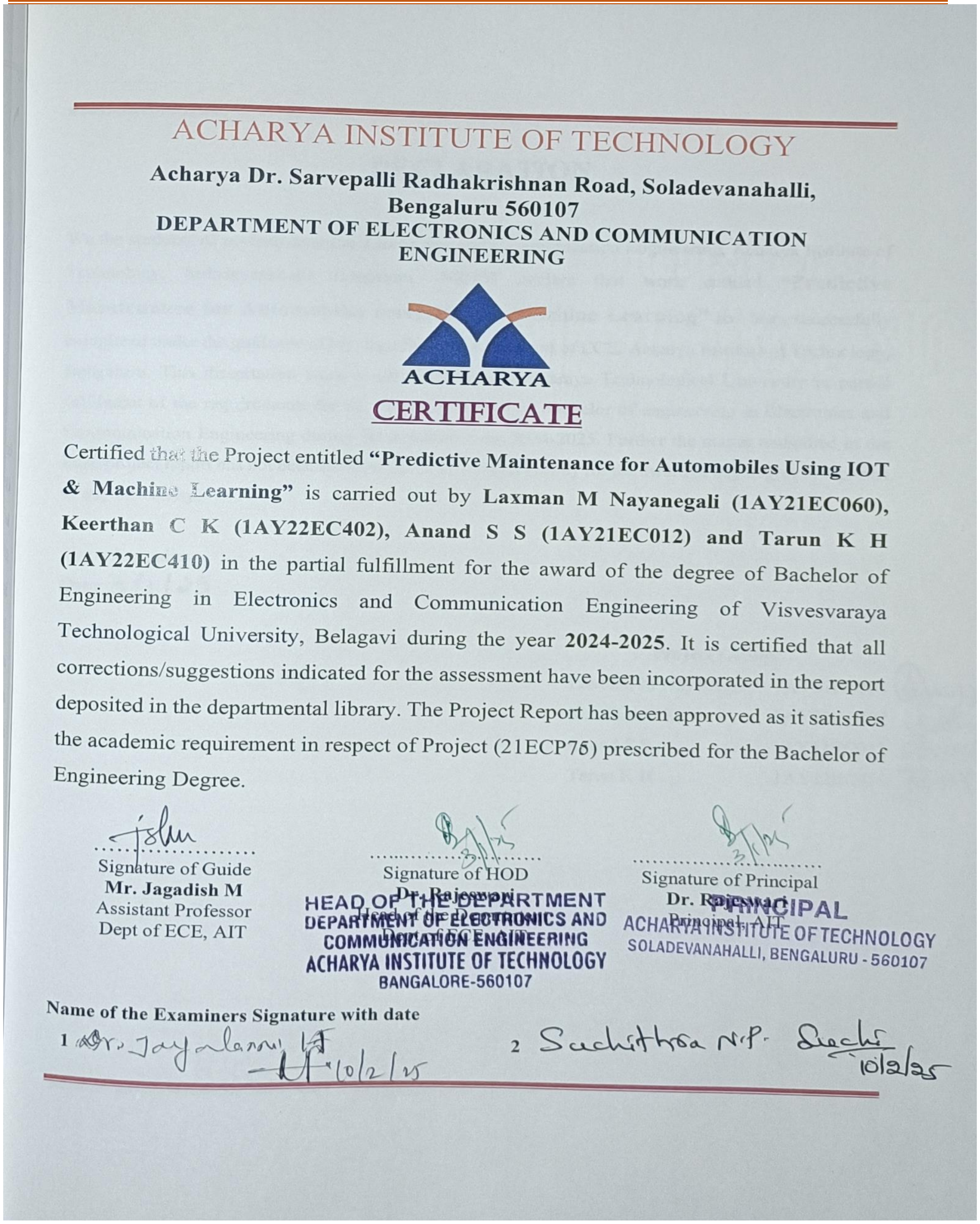
Asst Prof. Dept of ECE AIT

2024-2025

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Acharya Institute of Technology

Acharya Dr. Sarvepalli Radhakrishnan Road, Soladevanahalli, Bengaluru-56010



ACHARYA INSTITUTE OF TECHNOLOGY

Acharya Dr. Sarvepalli Radhakrishnan Road, Soladevanahalli, Bengaluru 560107

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ACHARYA

CERTIFICATE

Certified that the Project entitled **"Predictive Maintenance for Automobiles Using IOT & Machine Learning"** is carried out by **Laxman M Nayanegali (1AY21EC060), Keerthan C K (1AY22EC402), Anand S S (1AY21EC012) and Tarun K H (1AY22EC410)** in the partial fulfillment for the award of the degree of Bachelor of Engineering in Electronics and Communication Engineering of Visvesvaraya Technological University, Belagavi during the year **2024-2025**. It is certified that all corrections/suggestions indicated for the assessment have been incorporated in the report deposited in the departmental library. The Project Report has been approved as it satisfies the academic requirement in respect of Project (21ECP75) prescribed for the Bachelor of Engineering Degree.

[illegible]	[illegible] 3/1/25	[illegible] 3/1/25
Signature of Guide	Signature of HOD	Signature of Principal
Mr. Jagadish M	**Dr. Rajeswari**	**Dr. Rajeswari**
Assistant Professor	Head of the Department	Principal, AIT
Dept of ECE, AIT	Dept of ECE, AIT	

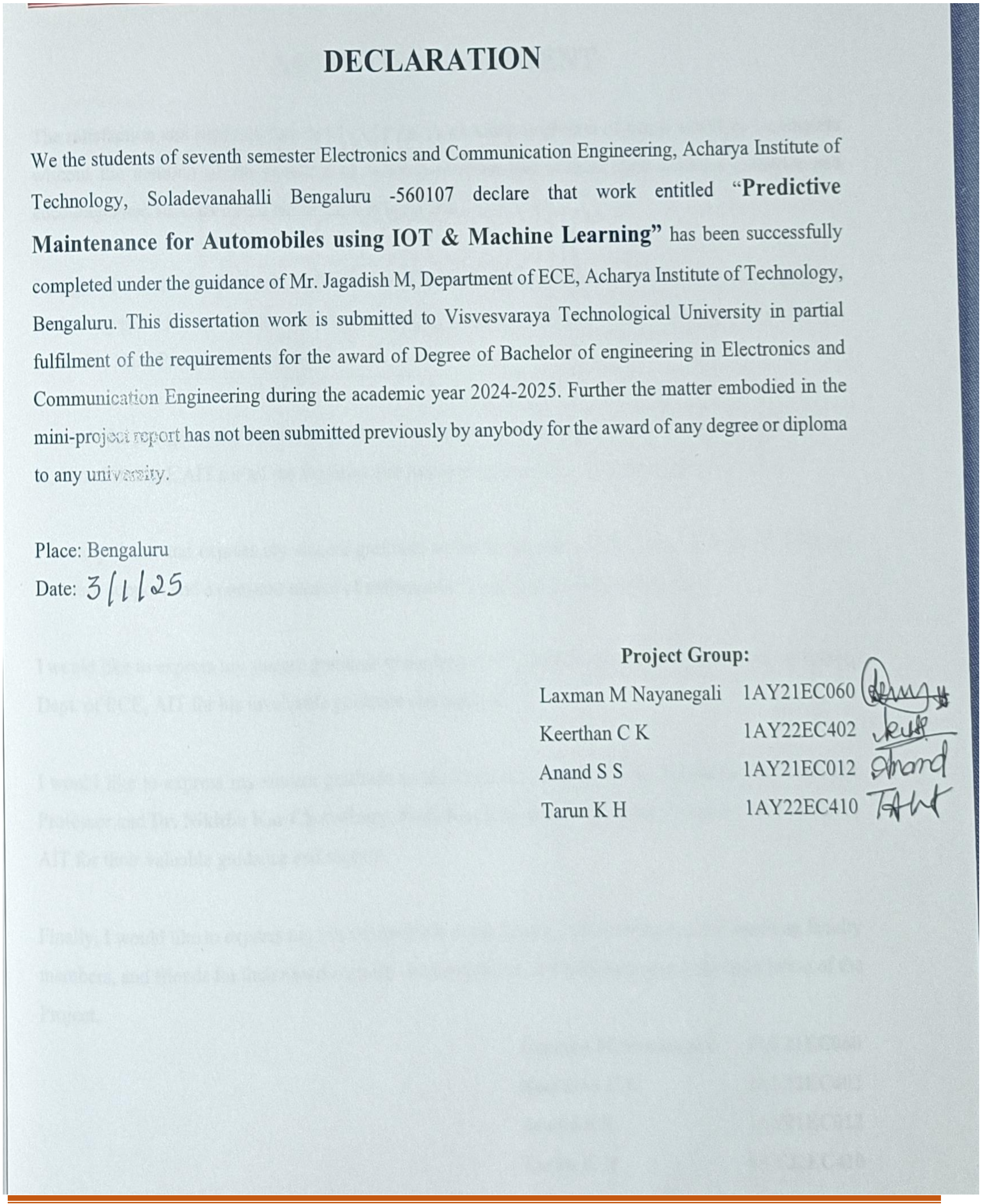
HEAD OF THE DEPARTMENT
DEPARTMENT OF ELECTRONICS AND
COMMUNICATION ENGINEERING
ACHARYA INSTITUTE OF TECHNOLOGY
BANGALORE-560107

PRINCIPAL
ACHARYA INSTITUTE OF TECHNOLOGY
SOLADEVANAHALLI, BENGALURU - 560107

Name of the Examiners Signature with date

1 Dr. Jayalaxmi H [illegible] 10/2/25

2 Suchithra N.P. [illegible] 10/2/25



DECLARATION

We the students of seventh semester Electronics and Communication Engineering, Acharya Institute of Technology, Soladevanahalli Bengaluru -560107 declare that work entitled **"Predictive Maintenance for Automobiles using IOT & Machine Learning"** has been successfully completed under the guidance of Mr. Jagadish M, Department of ECE, Acharya Institute of Technology, Bengaluru. This dissertation work is submitted to Visvesvaraya Technological University in partial fulfilment of the requirements for the award of Degree of Bachelor of engineering in Electronics and Communication Engineering during the academic year 2024-2025. Further the matter embodied in the mini-project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place: Bengaluru
Date: 3/1/25

Project Group:

Laxman M Nayanegali	1AY21EC060	[illegible]
Keerthan C K	1AY22EC402	[illegible]
Anand S S	1AY21EC012	Anand
Tarun K H	1AY22EC410	[illegible]

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of a task would be incomplete without the mention of the people who made it possible and without their constant guidance and encouragement success would not have been possible.

I am grateful to the institute **Acharya Institute of Technology management** for its ideas and inspiration for having provided us with the good infrastructure, laboratory, facilities, and inspiring staff which has made this Mini-Project successfully.

I would like to express my sincere gratitude to **Dr. Rajeswari** Principal, AIT and **Prof. Mari Gowda C K**, Vice Principal, AIT for all the facilities that has been extended by them throughout my work.

I heartily thank and express my sincere gratitude to **Dr. Rajeswari**, HOD, Dept. of ECE, AIT for her valuable support and a constant source of enthusiastic inspiration to steer us forward.

I would like to express my sincere gratitude to the Internal Guide **Mr. Jagadish M,** Assistant Professor, Dept. of ECE, AIT for his invaluable guidance and support.

I would like to express my sincere gratitude to the Project Coordinators **Dr. Jayalaxmi H,** Associate Professor and **Dr. Nikitha Kar Chowdhary, Prof. Kiran Kumar T** Assistant Professor, Dept. of ECE, AIT for their valuable guidance and support.

Finally, I would like to express my sincere gratitude to my parents, all teaching and non-teaching faculty members, and friends for their moral support, encouragement, and help throughout the completion of the Project.

Laxman M Nayanegali	**1AY21EC060**
Keerthan C K	**1AY22EC402**
Anand S S	**1AY21EC012**
Tarun K H	**1AY22EC410**

ABSTRACT

The complexity and hazards of automobiles and electrical Vehicles have posed a significant challenge in vehicle maintenance. A Proactive automobile maintenance approach is important for the reduction of surprise failures, reduction of downtime, and improvement in vehicle performance. There is a need for a IoT-based predictive maintenance solution for key components of vehicles, where the information from sensors in real-time about the status of temperature, vibration, battery conditions, among other parameters, can predict failures with necessary time and action for maintenance based on machine learning algorithms and data analytics. This project uses the advances in the related technologies of automobile vehicle Predictive maintenance and machine learning algorithm to address the inadequacies of current maintenance methods by developing a proactive solution for safe, cost-effective, and overall reliability of the vehicle. The expected output will be a real-time dashboard, an alert system, and validated predictive algorithms that hugely improve vehicle performance and operational cost.

Table of Contents

Table of Figures

CHAPTER: 01

INTRODUCTION

Predictive maintenance (PDM) in the automotive industry is a great example of predictive analytics. It helps businesses determine when a machine or vehicle part needs servicing, using techniques such as data mining, data preprocessing and employing machine learning algorithms. Predictive maintenance uses historical and real-time data from various parts of your operation to anticipate problems before they happen. Some examples of using predictive maintenance and predictive maintenance sensors include vibration analysis, oil analysis, thermal imaging and equipment observation. In an automotive manufacturing environment, if there is scheduled or unscheduled downtime, the corresponding costs may result in a serious setback. With predictive maintenance, it is possible to constantly monitor the health of industrial equipment in real time and predict the probability of failures. This improves the efficiency of operations and reduces the maintenance cost of equipment. [1]

Traditional vehicle maintenance methods, such as periodic maintenance and reactive repairs, often lead to unnecessary part replacements, unexpected breakdowns, and high operational costs for vehicle owners. This project focuses on implementing predictive maintenance for automotive systems using IoT. By leveraging sensor data and real-time monitoring, the system predicts potential failures, reducing downtime and maintenance costs, while improving vehicle performance and safety. Unplanned vehicle breakdowns lead to costly repairs, unexpected downtime, and safety risks. Present maintenance methods are ineffective because they rely totally on servicing that is scheduled or based on break-down/reactive events. There is a growing need for an intelligent, real-time solution that predicts failures before they occur, ensuring optimal performance, safety, and cost-efficiency in automotive systems. Predictive maintenance using IoT technology is transforming the automotive industry by providing real time insights into vehicle health and performance. Traditional maintenance methods rely on scheduled servicing or fixing issues after they arise, which can lead to unexpected breakdowns and increased repair costs. In contrast, predictive maintenance leverages data collected from sensors installed in the vehicle to monitor its condition continuously and predict potential failures before they happen.

The use of IoT enables seamless connectivity between the vehicle’s onboard sensors and machine learning model. This connected network allows the collection and analysis of critical data such as engine temperature, fuel levels, tire pressure, and battery health. By continuously monitoring parameters such as temperature, vibration, oil level, battery performance, and gas levels, the system identifies potential failures before they occur. The integration of an ESP32 microcontroller with a Python Flask-based server enables real-time data processing, anomaly detection, and user-friendly dashboard visualization.

CHAPTER: 02

LITERATURE SURVEY

Sl.no	Author(s)	Title	Publication	Year	Key contributions
1	Arena, F.; Collotta, M.; Luca, L.; Ruggieri, M.; Termine, F.G.	Predictive Maintenance in the Automotive Sector: A Literature Review	Mathematical and Computational Applications MDPI.	2021	A systematic literature review of statistical inference approaches, and ML techniques for predictive maintenance in the automotive sector.
2	Rohit Dhall, Vijender Solanki	An IoT Based Predictive Connected Car Maintenance Approach	International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 4, Nº3.	2022	PDM for connected cars by using real-time data to monitor vehicle health and predict issues before they occur.
3	Jiang, R., et al.	Predictive maintenance of autonomous vehicles using machine learning and IoT	Sustainable Computing: Informatics and Systems.	2021	PDM using both supervised techniques.
4	Zhang, Y., Li, H., & Tang, X.	Machine learning in autonomous vehicles: A survey.	IEEE Transactions on Intelligent Transportation Systems.	2020	Machine learning algorithms with PDM for automobiles.
5	Sherien N. Elkateb, Ahmed Metwalli, Abdelrahman Shendhy Ahmed E. B. Abu-Elanien	Machine learning and IoT – Based predictive maintenance approach for industrial applications	Alexandria Engineering Journal	2024	Hyperparameter tuning and cross- validation techniques

Figure 2.1: Literature Survey

CHAPTER: 03

OBJECTIVES

3.1 Problem Statement

Traditional vehicle maintenance methods are inefficient, leading to:

- Unnecessary component replacements.
- High operational costs.
- Unplanned downtime.

Modern vehicles lack real-time monitoring and predictive capabilities for critical components like the engine, battery, and brakes. This results in difficulty in identifying potential failures early, suboptimal performance and reliability.

3.2 Objectives

1. Develop a real-time vehicle monitoring system using ESP32 and various sensors.
2. Predict vehicle maintenance requirements using machine learning algorithms.
3. Send alerts to vehicle owners via Telegram messages for proactive maintenance.
4. Detect accidents and other critical issues such as overheating, low oil levels, and poor battery health.
5. Ensure wireless communication of sensor data to a central laptop for analysis and storage.

3.3 Proposed System

The proposed system introduces a predictive maintenance framework that combines IoT and machine learning for real-time vehicle monitoring and intelligent analysis. Key features of the proposed system include: [2]

1. **Sensors Integration**:
   - **DHT11**: Monitors engine temperature.
   - **DS18B20**: Tracks battery temperature.
   - **Voltage and Current Sensors**: Measure battery health.
   - **Ultrasonic Sensor**: Detects engine oil levels.
   - **MQ-3 Smoke Sensor**: Monitors smoke levels.
   - **ADXL345 Accelerometer**: Detects vibrations and accidents.
2. **Data Transmission**:
   - Sensor data is wirelessly transmitted to a central laptop via Zigbee modules.
3. **Predictive Analysis**:
   - Machine learning algorithms analyze sensor data to predict maintenance needs and vehicle condition.
4. **User Alerts**:
   - Maintenance alerts (e.g., low oil, overheating, poor battery health) are sent to users via Telegram messages.
5. **Accident Detection**:
   - The system detects accidents based on vibration data and sends immediate alerts.

CHAPTER : 04

HARDWARE DESCRIPTION

4.1 ESP 32 Microcontroller

ESP32 is a single chip 2.4 GHz Wi-Fi and Bluetooth combo chip designed with TSMC ultra-low power 40 nm technology. It is designed and optimized for the best power performance, RF performance, robustness, versatility, features and reliability, for a wide variety of applications, and different power profiles.

ESP32 is designed for mobile, wearable electronics, and Internet of Things (IoT) applications. It has many features of the state-of-the-art low power chips, including fine resolution clock gating, power modes, and dynamic power scaling. For instance, in a low-power IoT sensor hub application scenario, ESP32 is woken up periodically and only when a specified condition is detected; low duty cycle is used to minimize the amount of energy that the chip expends. The output power of the power amplifier is also adjustable to achieve an optimal tradeoff between communication range, data rate and power consumption. [3]

ESP32 is the most integrated solution for Wi-Fi + Bluetooth applications in the industry with less than 10 external components. ESP32 integrates the antenna switch, RF balun, power amplifier, low noise receives amplifier, filters, and power management modules. As such, the entire solution occupies minimal Printed Circuit Board (PCB) area. ESP32 uses CMOS for single-chip fully-integrated radio and baseband, and also integrates advanced calibration circuitries that allow the solution to dynamically adjust itself to remove external circuit imperfections or adjust to changes in external conditions. As such, the mass production of ESP32 solutions does not require expensive and specialized Wi-Fi test equipment.

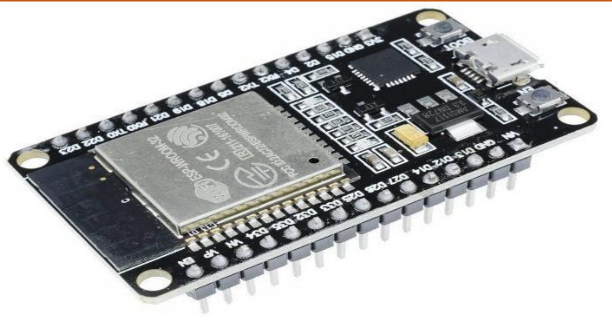


Figure 4.1: ESP 32 Microcontroller

4.2 Regulated power supply:

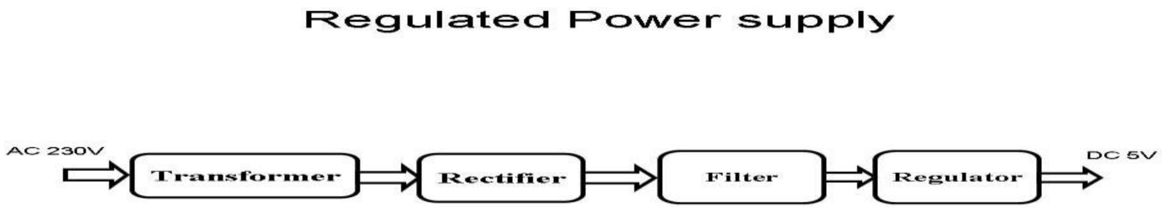


Figure 4.2: Regulated power supply

A transformer is an electrical device that transfers energy between circuits using inductively coupled conductors while keeping the same frequency. It works on the principle of mutual induction, where a changing magnetic field created by the primary winding induces a voltage in the secondary winding. This principle allows energy to move from the primary circuit to the secondary circuit. When a load is connected, current flows through the secondary winding. The alternating current (AC) magnetic field, which fluctuates at 50 Hz for AC mains, interacts with nearby coils, inducing voltage in them. This basic concept is what makes transformers function. [4]

A rectifier is responsible for converting AC to direct current (DC) through a process called rectification, which is crucial for power supplies and signal detection. It typically employs multiple diodes to achieve efficient AC-to-DC conversion, although earlier models used vacuum tubes or selenium stacks. Filters are used to smooth out pulsating DC by eliminating unwanted frequencies, while voltage regulators ensure that fluctuating input voltage is maintained at a constant output. Regulators like the LM78XX series for positive input and LM79XX for negative input are reliable devices that protect against over-current or overheating. However, applying reverse voltage can lead to immediate damage.

4.3 Voltage Sensor:

A voltage sensor is an essential electronic component designed to measure and monitor electrical potential difference in various electronic and electrical systems. It typically consists of a voltage divider circuit that allows microcontrollers like ESP32 to safely measure higher voltage levels by scaling down the input voltage to a range compatible with the analog input pins. These sensors are crucial in battery monitoring applications, enabling precise measurement of battery charge levels, detecting voltage drops, and providing real-time status updates. They use resistor networks to divide the input voltage, ensuring safe and accurate voltage readings while protecting the microcontroller from potential damage caused by high voltage inputs.

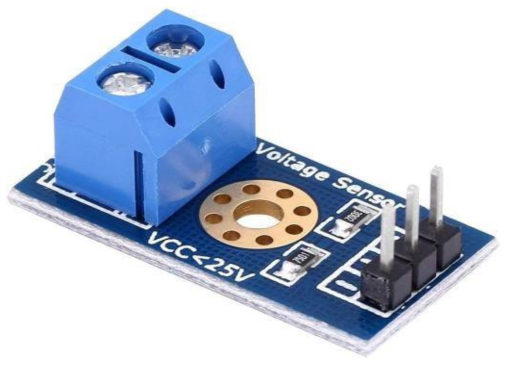


Figure 4.3: Voltage Sensor

Sensors are basically a device which can sense or identify and react to certain types of electrical or some optical signals. This sensor mainly includes voltage divider circuit. The resistor in the circuit works as a sensing element. The voltage can be separated into two resistors like a reference voltage & variable resistor to make a circuit of the voltage divider. [5]

4.4 LCD Display 16X2

The term LCD stands for liquid crystal display. It is one kind of electronic display module used in an extensive range of applications like various circuits & devices like any prototype, circuits, mobile phones, calculators, computers, TV sets, etc. These displays are mainly preferred for multi-segment light-emitting diodes and seven segments. The main benefits of using this module are inexpensive; simply programmable, animations, and there are no limitations for displaying custom characters, special and even animations, etc.

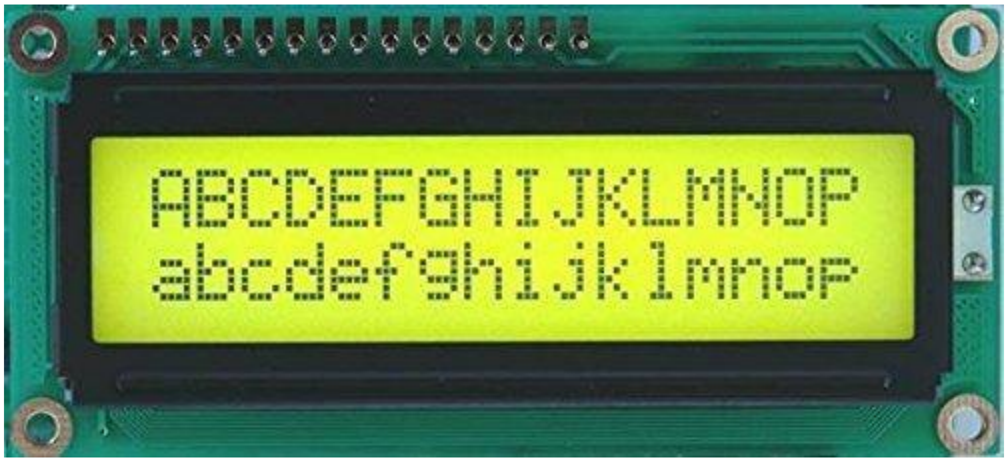


Figure 4.4: LCD Display

4.4.1 Registers of LCD

A 16×2 LCD has two registers like data register and command register. The RS (register select) is mainly used to change from one register to another. When the register set is ‘0’, then it is known as command register. Similarly, when the register set is ‘1’, then it is known as data register.

4.4.2 Command Register

The main function of the command register is to store the instructions of command which are given to the display. So that predefined tasks can be performed such as clearing the display, initializing, set the cursor place, and display control. Here commands processing can occur within the register.

- **Data Register**

The main function of the data register is to store the information which is to be exhibited on the LCD screen. Here, the ASCII value of the character is the information which is to be exhibited on the screen of LCD. Whenever we send the information to LCD, it transmits to the data register, and then the process will be starting there. When register set =1, then the data register will be selected.

4.5 Current Sensor Module

The module is designed using the ZMCT series of small size high-precision micro-CT and high-precision operational amplifier circuits for more accurate sampling and proper signal compensation. It is best solution for the signal acquisition of AC current within 5A range.

The corresponding output voltage Analog AC signal can be adjusted using the potentiometer. You can adjust the amplification ratio and the amplification range (0 -100 times), but the max voltage at the output will not more than half of VCC applied voltage.

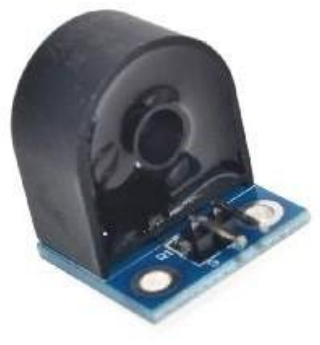


Figure 4.5: Current Sensor Module

ZMCT103C AC current Sensor is the best for the purpose of the DIY project and industrial application, where we need to measure the accurate AC current with current transformer. This is a perfect choice to measure the AC current using Arduino/ESP8266/Raspberry Pi like an opensource platform. In many electrical projects, engineer directly deals with measurements with few basic requirements like

- High galvanic isolation
- High accuracy
- Good Consistency

This is a high precision micro current Transformer. This module makes it easy to monitor AC mains current up to 5 Amps. ZMCT103 is a PCB mount current transformer with 1000:1 turns ratio and Dimensions: 28 x 12 x 15 mm (L*W*H).

4.6 Temperature and Humidity Sensor

This DHT11 temperature and humidity sensor comes in a very small package and can be found very easily in online & offline market as it is very popular among beginners and hobbyist. It can easily be connected to any microcontroller available in the market like Arduino & Raspberry Pi. Only some libraries need to be installed as the data is retrieved from the single output.

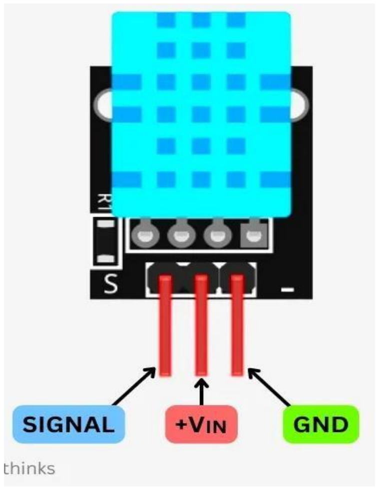


Figure 4.6: DHT 11 Sensors

DHT11 sensor has four pins you can see in the above dht11 pin diagram. There are 4 pins. VCC, GND, and a data pin. VCC and ground are to be connected to the power. From these two pins, you have to give power to the dht11 sensor. And there is one data pin from which you can extract the data. From this single pin, you can extract both humidity and temperature. But first, you have to install the library to make the code. Don’t give more than 5v from VSS pin. And gnd pin always connects to the ground pin of the power source. [6]

The working of this **DHT11 temperature and humidity sensor** is very simple for all the models available in the market. As the data is given out via a single pin. So adequate delay must be given between retrieving two consecutive values. The 4th pin on the module is the NC pin, which is not connected generally to any pin of the microcontroller. But you can use this to provide structure strength to the module, as it is rather compact.

4.7 Current Sensor:

A current sensor is a sophisticated electronic device designed to measure electrical current flow in a circuit by converting the current into a proportional voltage signal that can be easily read by microcontrollers. Typically, these sensors use various technologies such as Hall effect, shunt resistors, or current transformers to measure electrical current without directly interrupting the circuit. In IoT and electrical monitoring applications, current sensors play a critical role in measuring power consumption, detecting overload conditions, and monitoring electrical system performance. They provide real-time insights into energy usage, enable predictive maintenance, and support advanced power management strategies across diverse applications including renewable energy systems, industrial equipment, and electric vehicle charging infrastructures.

Figure 4.7: Current Sensor

4.8 Smoke MQ-3 Sensor:

The MQ-3 is a highly sensitive semiconductor-based gas sensor specifically designed for detecting combustible gases and smoke in various environments. It can detect multiple gases including liquefied petroleum gas (LPG), propane, hydrogen, methane, and alcohol, making it versatile for safety and detection applications. The sensor operates by measuring changes in electrical resistance when exposed to combustible gases, with sensitivity that can be adjusted using a potentiometer. Its analog output allows microcontrollers to interpret gas concentration levels, triggering alerts or activating safety mechanisms when dangerous gas levels are detected. Commonly used in fire alarm systems, gas leak detectors, and industrial safety applications, the MQ-3 provides an affordable and reliable solution for gas and smoke detection across home, commercial, and industrial settings.

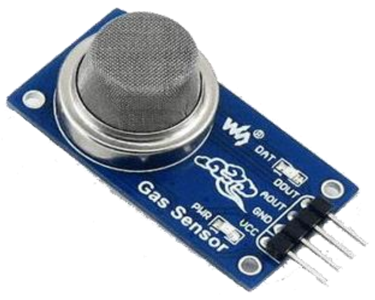


Figure 4.8: MQ-3 Sensor

4.9 DS18B20

is a highly versatile and sophisticated digital temperature sensor renowned for its precision and simplicity in temperature measurement applications. Manufactured by Dallas Semiconductor (now part of Maxim Integrated), this one-wire digital thermometer provides accurate temperature readings with a remarkable temperature range of -55°C to +125°C and an impressive accuracy of ±0.5°C between -10°C and +85°C.

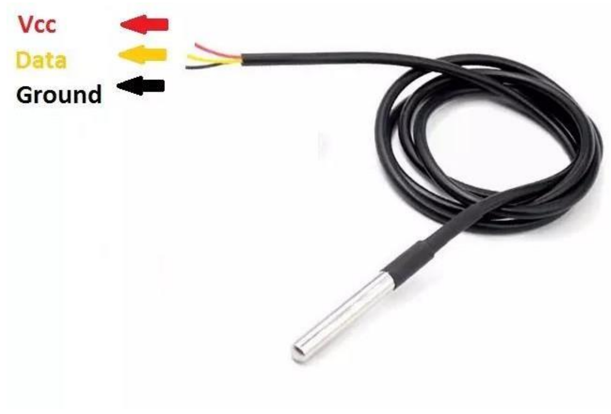


Figure 4.9: DS18B20

Technical Operation: The DS18B20 communicates using the 1-Wire protocol, which allows multiple sensors to be connected to a single data line. Each sensor has a unique 64-bit ROM code, enabling multiple sensors to coexist on the same bus without address conflicts. The sensor can provide temperature readings directly in digital format, eliminating the need for complex analog-to-digital conversion.

Communication Protocol: Sensors are connected via a single data wire, which also provides power in parasitic mode. The microcontroller (like ESP32) initiates communication by sending specific commands to read temperature, making it extremely efficient and easy to integrate into various projects. Practical Implementation: When used with microcontrollers like ESP32, the DS18B20 requires minimal external components. A 4.7kΩ pull-up resistor is typically used between the data line and power supply to ensure proper communication. The sensor can be directly connected to digital pins, making it incredibly user-friendly for hobbyists and professional developers alike. The DS18B20's combination of accuracy, versatility, and ease of use makes it a preferred choice for projects requiring reliable temperature measurements across various domains.

4.10 US Sensor

The HC-SR04 ultrasonic sensor uses SONAR to determine the distance of an object just like the bats do.

It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package from 2 cm to 400 cm or 1” to 13 feet.

The operation is not affected by sunlight or black material, although acoustically, soft materials like cloth can be difficult to detect. It comes complete with ultrasonic transmitter and receiver module.

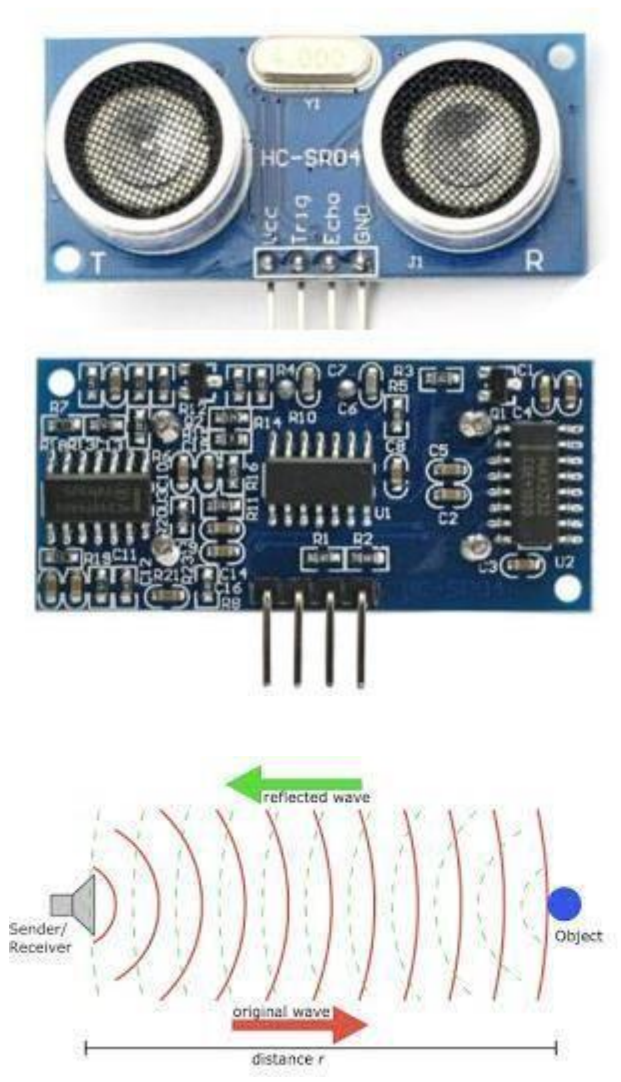


Figure 5.10: Ultra Sonic Sensor

4.11 Jumper Wires

A jump wire (also known as jumper, jumper wire, jumper cable, DuPont wire, or DuPont cable) is an electrical wire or group of them in a cable with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering.

Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

Vibration Sensor Overview: A vibration sensor, also known as an accelerometer or vibration module, is an electronic device designed to detect and measure mechanical vibrations, acceleration, and movement in various applications. These sensors convert mechanical motion into electrical signals, providing critical data about vibration characteristics, intensity, and frequency.

4.12 ADXL Sensor

An accelerometer is a tool that measures the vibration, motion, or acceleration of a structure. Cameras and smartphones these days use an accelerometer consisting of an axis-based motion sensor. It is an electromechanical device that measures either static or dynamic acceleration. Acceleration, as we know, is the measure of change in velocity upon a given time.

Accelerometers are used in the compass app you use on your phone. The motion sensors in accelerometers can detect earthquakes too. Another example is when the accelerometers measure the gravitational pull to determine at which angle is the device being titled.

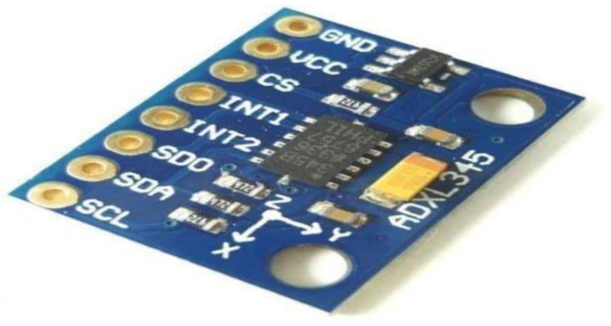


Figure 5.11: ADXL345 Sensor

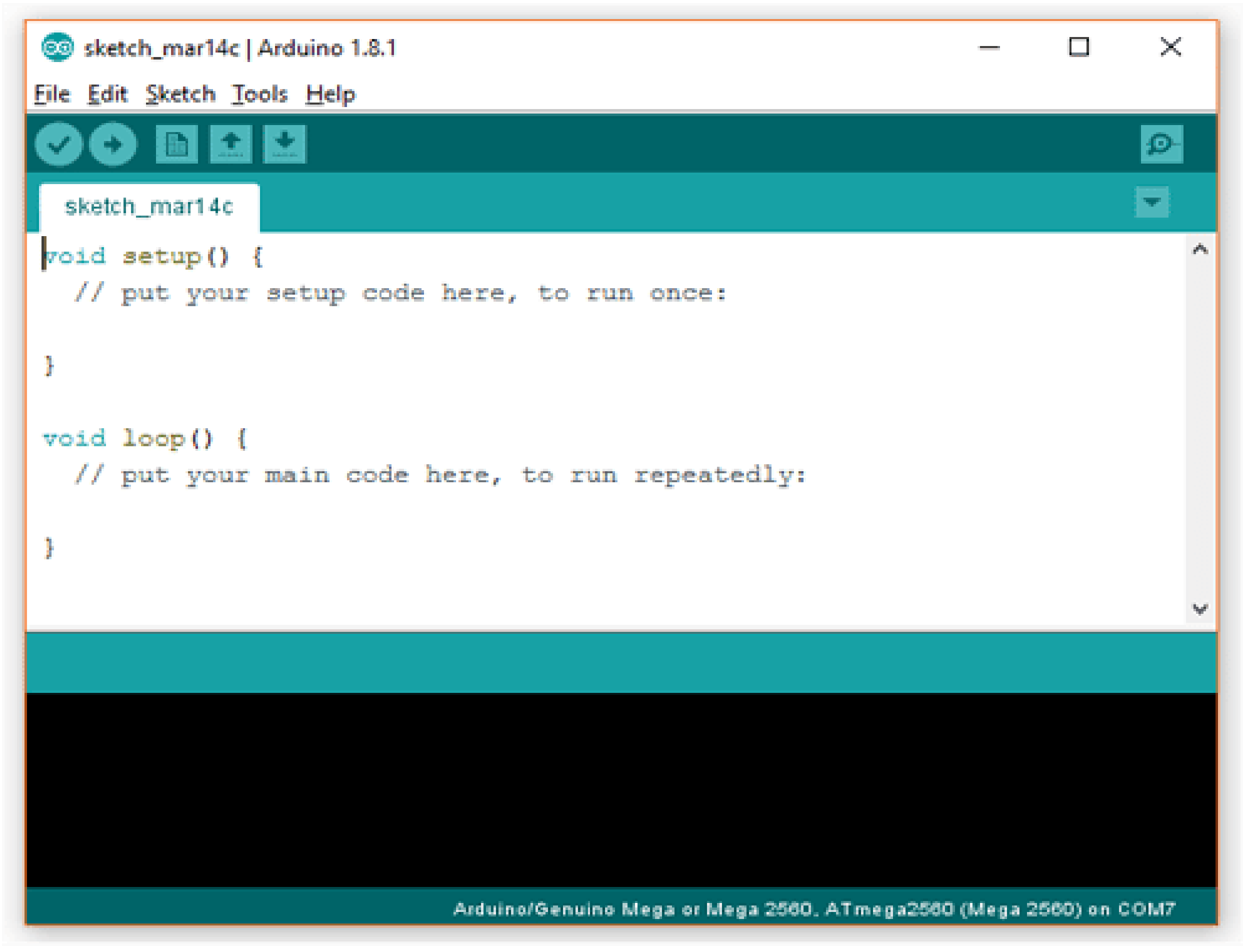
Accelerometer is an electromechanical device that measures the force of acceleration due to gravity in a g unit. On the earth, 1g means an acceleration of 9.8 m/s2 is present. On the moon, it is 1/6th of earth and on mars, it is 1/3rd of earth. Accelerometer can be used for tilt-sensing applications as well as dynamic acceleration resulting from motion, shock, or vibration.

CHAPTER: 05

SOFTWARE DESCRIPTION

5.1 Introduction to Arduino IDE

The Arduino IDE is incredibly minimalistic, yet it provides a near-complete environment for most Arduino-based projects. The middle section of the IDE is a simple text editor that where you can enter the program code. The bottom section of the IDE is dedicated to an output window that is used to see the status of the compilation, how much memory has been used, any errors that were found in the program,



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM7

Figure 5.1: Arduino IDE Editor Window

Projects made using the Arduino are called sketches, and such sketches are usually written in a cut-down version of C++ (a number of C++ features are not included). Because programming a microcontroller is somewhat different from programming a computer, there are a number of device-specific libraries (e.g., changing pin modes, output data on pins, reading analog values, and timers).

5.2 Flask Framework

Flask is a lightweight and flexible web framework created in Python, aimed at developing web applications and APIs. As a microframework, it offers just the core tools and depends on extensions for extra features, which makes it very versatile for different applications. Flask is straightforward to learn and use, making it perfect for IoT and machine learning integration projects like ours. [7]

- **Features:**

Figure 5.2: Flask Framework

- **WERKZEUG WSGI Toolkit**

WERKZEUG (German for "tool") is a comprehensive WSGI (Web Server Gateway Interface) library that acts as the foundation for Flask. It handles HTTP requests and responses, routing, and error management. In Our project it manages incoming sensor data sent from ESP32 via HTTP requests & ensures reliable routing of data to appropriate endpoints for processing.

- **Jinja2 Template Engine**

Jinja2 is a powerful templating engine used by Flask to generate dynamic HTML pages. It allows developers to embed Python code directly into HTML templates for seamless integration of back-end logic with the front-end interface. It dynamically renders sensor data and ML predictions on the web dashboard, The Displayed alerts and historical trends retrieved from the SQLite database.

5.3 SQ Lite

SQLite is a lightweight, self-contained, and serverless relational database management system. It's commonly used in applications that prioritize simplicity, portability, and minimal setup. Unlike traditional databases, SQLite works directly with a single file, which makes it very efficient and easy to incorporate into embedded systems or small-scale projects like ours. [8]

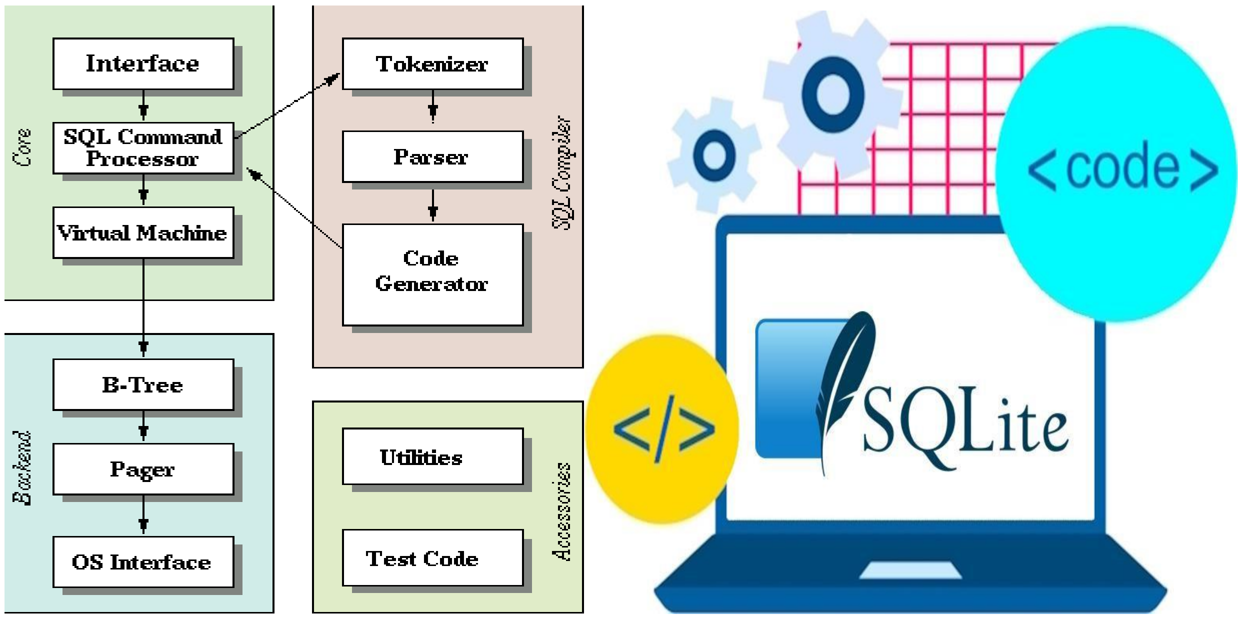


Figure 5.3: SQLite

Features:

- Serverless: Functions without a dedicated server, minimizing resource needs.
- Self-Contained: All database logic is encapsulated in a single library, eliminating the need for extra dependencies.
- Zero Configuration: No setup, installation, or configuration is required, making it perfect for rapid prototyping.
- Cross-Platform: Operates smoothly across various platforms, including Windows, Linux, and embedded systems.
- Lightweight: Has a small storage footprint (~250KB), making it ideal for resource-limited systems like IoT devices.

5.4 Machine Learning Model

Machine learning is crucial in the predictive maintenance system, as it analyzes sensor data to identify patterns that can forecast failures before they happen. In this project, the Random Forest Algorithm serves

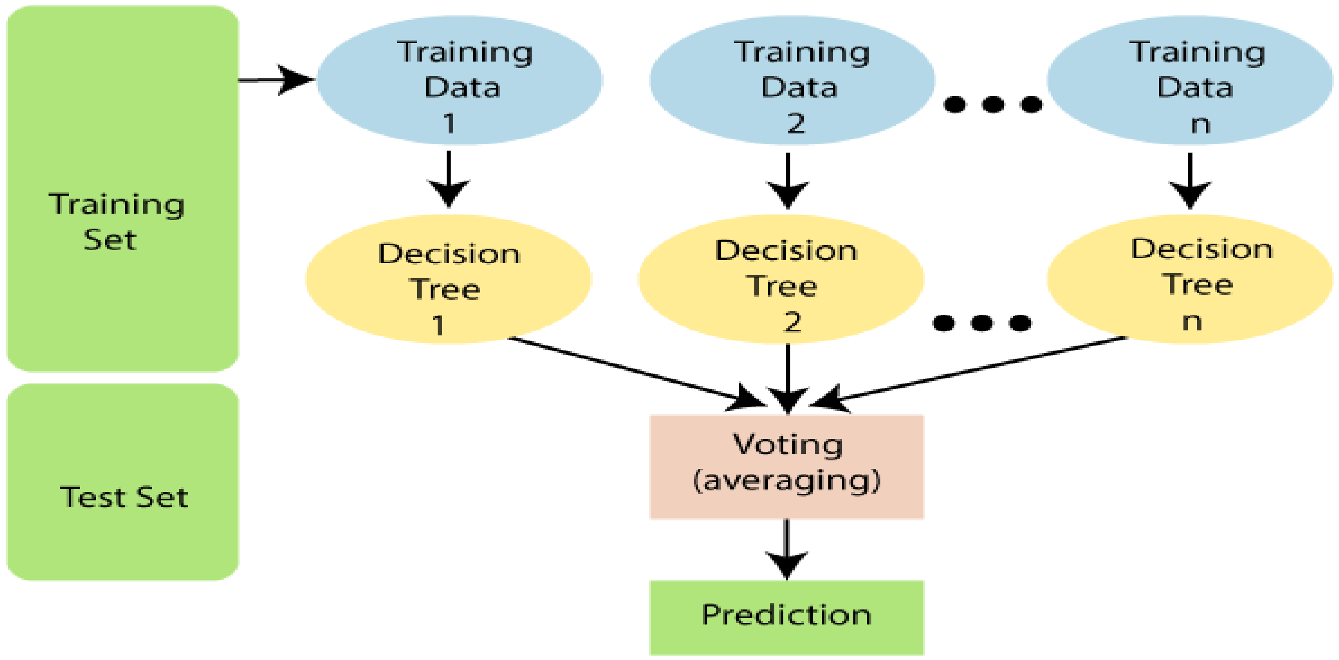


Figure 5.4: Diagram of Machine Learning Model

as the primary machine learning model because of its high accuracy, robustness, and capability to manage both classification tasks (like anomaly detection) and regression tasks (such as predicting Remaining Useful Life). By utilizing historical data, the model facilitates proactive interventions, minimizing downtime and improving vehicle reliability. [9]

Features:

- Anomaly Detection: Detects unusual patterns in sensor data that may signal potential failures. Remaining Useful Life (RUL)
- Prediction: Predicts how much longer vehicle components will last based on sensor data.
- Robustness: Effectively manages noisy or incomplete data, providing dependable predictions.
- Scalability: Can be expanded with more data for wider applications across various components.
- Versatility: Facilitates both real-time predictions and offline analysis of historical data.
- Analysis: Determines which sensors are most influential in predictions, enhancing system efficiency.

5.5 JavaScript

JavaScript is a powerful, dynamic scripting language used to create interactive effects within web browsers. It's an essential tool for web development, enabling user interaction without needing to reload the page. JavaScript is widely used in modern web development to create dynamic, client-side applications.

Applications:

- Data Visualization
- User Interaction
- Integration with IoT

- Machine Learning Predictions

5.6 CSS (Cascading Style Sheets)

CSS is a stylesheet language used to describe the presentation of a web page written in HTML or XML. It controls the layout, design, colors, and fonts of the content displayed on web pages.

Applications:

- Dashboard Design
- Responsive Design
- Real-time Data Display
- Theme Customization

5.7 HTML (Hypertext Markup Language)

HTML is the standard markup language used to create web pages. It forms the backbone of web content, structuring the content and linking it to other resources like stylesheets (CSS) and scripts (JavaScript).

Applications:

- Web Interface
- Sensor Data Display
- User Input Forms

Figure 5.5: HTML, CSS, Java Script

5.8 Telegram Bot Integration

The Telegram bot plays a crucial role in the project by sending real-time notifications to users regarding any anomalies found in the vehicle's critical components. By utilizing Telegram's secure and dependable API, the bot guarantees that users receive alerts straight to their mobile devices, which improves accessibility and user engagement. It serves as a link between the system's back-end processes and user interactions, providing timely updates on sensor anomalies like low oil levels, overheating, or gas leaks. Its straightforward setup, ease of use, and mobile accessibility make it a valuable addition to the predictive maintenance system.

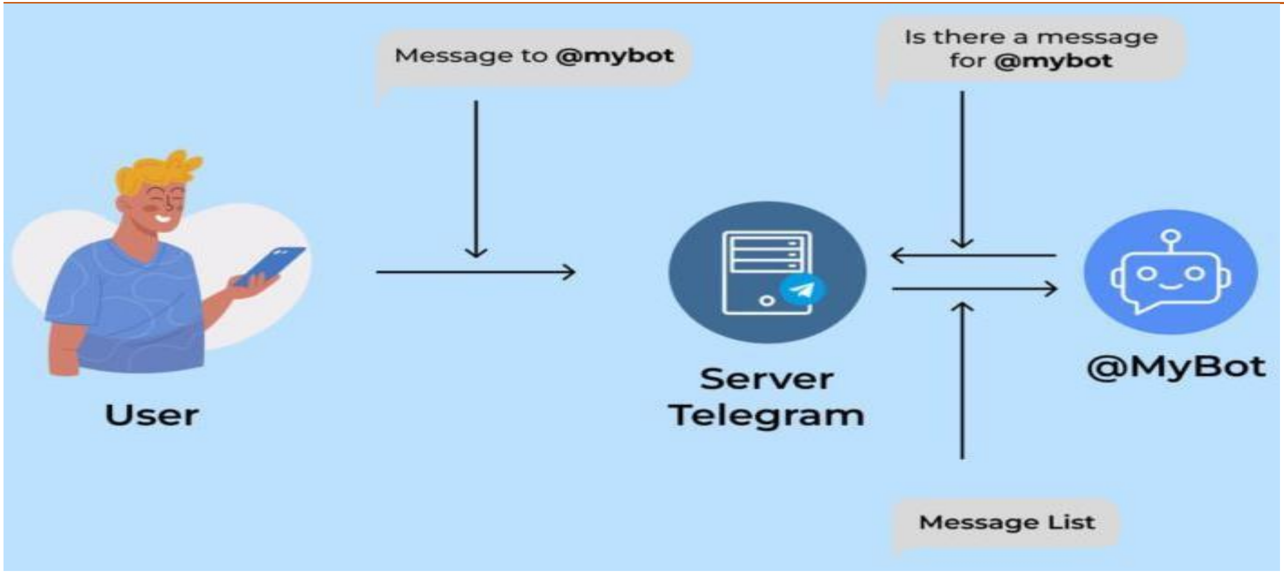


Figure 5.6: Telegram Bot

Benefits of Using Telegram Bot

1. Instant Notifications: Real-time alerts ensure timely action.
2. User-Friendly Interface: Users receive alerts directly on their phones.
3. Two-Way Communication: Allows users to query the system remotely.
4. Low Cost: Telegram API is free and efficient for IoT applications.

CHAPTER: 06

METHODOLOGY

6.1 System Design and Architecture

IoT Sensors: A variety of sensors are utilized to monitor the condition of automobile components in real-time. These include vibration sensors, temperature sensors, pressure sensors, and other relevant devices that can identify wear and tear in vehicle parts. The sensors are strategically placed on critical components of the vehicle, such as the engine, transmission, and wheels.

Data Transmission & Web Interface: The information gathered from the sensors is sent to a central server for processing. Communication protocols like MQTT or HTTP are employed to transmit sensor data to either a cloud or local server, where further analysis takes place. A user-friendly web interface is created to present real-time data, alerts, and maintenance predictions to users. This interface facilitates interaction with the system, allowing users to input data, view predictive maintenance results, and receive notifications regarding necessary maintenance tasks.

Machine Learning Model: At the heart of the predictive maintenance system are machine learning algorithms that analyze the data collected from the sensors to forecast potential failures or maintenance requirements. The Random Forest algorithm is used to forecast potential failures and maintenance requirements based on sensor data. This ensemble learning technique builds multiple decision trees during the training phase and provides the most common class (for classification) or the average prediction (for regression) as the output. Each decision tree is trained on a random subset of the data using bootstrapping, and at each split, a random selection of features is evaluated. This approach helps to minimize overfitting and improves the model's ability to generalize and models trained on historical data, enables them to recognize patterns that may indicate equipment failure. [10]

Alert Mechanism: In the predictive maintenance system multiple sensor module are used to provide timely and effective notifications. When the machine learning model detects a fault or anomaly, the system activates alerts through three main channels: an LCD display, a buzzer, and a Telegram bot. The LCD display offers real-time visual alerts on the vehicle's dashboard or in the maintenance area, showing messages such as “Engine Overheating” or “Vibration Anomaly Detected.” At the same time, the buzzer sounds an audible alarm to quickly capture the attention of the driver or mechanic. Furthermore, the Telegram bot sends personalized notifications to the user or maintenance team, enabling remote monitoring and interaction. These alerts help ensure that any potential issues are addressed promptly, minimizing the risk of unexpected breakdowns and prolonging the vehicle's lifespan.

6.2 Block diagram

The block diagram illustrates a vehicle predictive maintenance system that combines IoT sensors, data processing, and machine learning to maintain optimal performance. Sensors such as temperature, vibration, fuel vapor, and ultrasonic oil level are strategically positioned on key vehicle components to monitor conditions in real-time. The ESP32 microcontroller gathers sensor data and transmits it to a backend machine learning model using a Zigbee module. This model analyzes the data to identify faults, forecast maintenance requirements, and avert potential failures. Alerts are shown on an LCD, activated by a buzzer, and sent remotely through a Telegram bot, allowing users to respond promptly and prevent unexpected breakdowns. Additionally, the system includes a dashboard interface for real-time monitoring and insights.

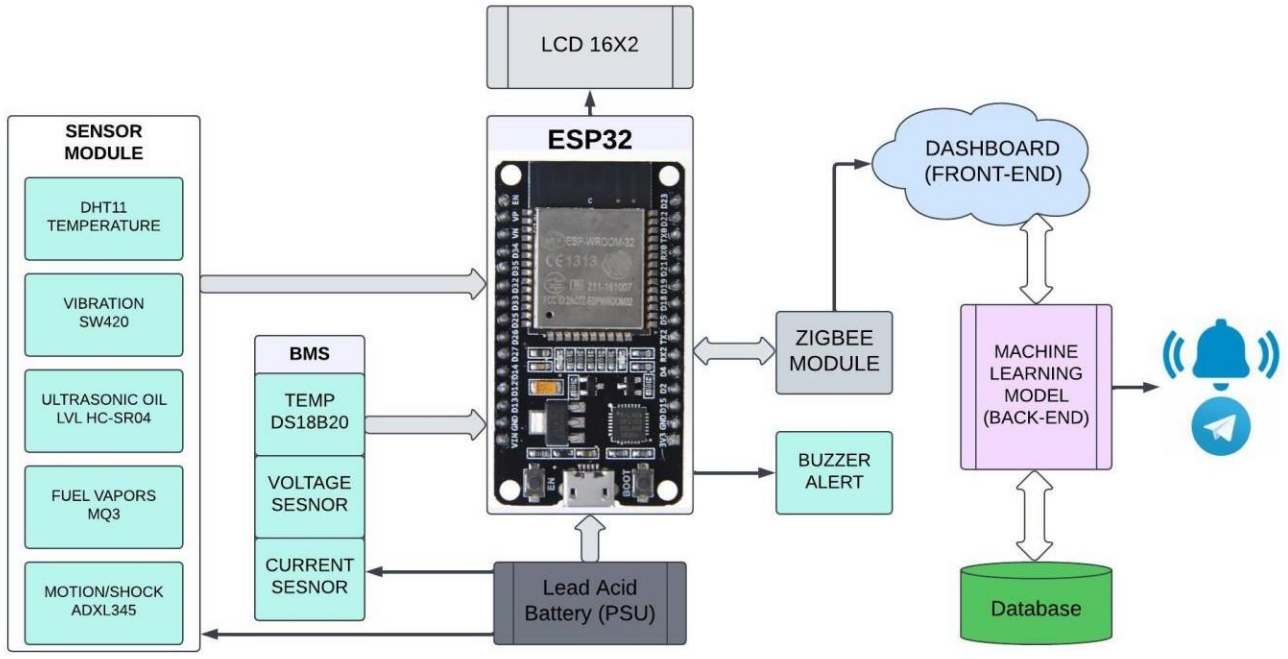


Figure 6.1: Block Diagram

The block diagram illustrates the architecture of a Predictive Maintenance System for Automobiles that utilizes IoT and machine learning. The components and their functions are detailed below:

1. Sensor Module

This module gathers real-time data from different parts of the vehicle:

- DHT11 (Temperature Sensor): Monitors the ambient temperature.

- Vibration Sensor (SW420): Detects unusual vibrations that could signal faults.

- Ultrasonic Sensor (HC-SR04): Measures the oil level to identify shortages.

- Fuel Vapors Sensor (MQ3): Detects fuel vapor leaks.

- Motion/Shock Sensor (ADXL345): Monitors shocks or jerks that may indicate physical impacts or system problems.

2. Battery Management System (BMS)

The BMS maintains the health of the vehicle’s power supply by monitoring:

- Temperature Sensor (DS18B20): Measures the battery’s temperature to detect overheating.

- Voltage Sensor: Keeps track of the battery voltage to ensure it operates correctly.

- Current Sensor: Monitors current usage to identify overloading or abnormal draw.

3. ESP32 Microcontroller

The ESP32 acts as the central controller:

- Collects data from the sensors and the BMS.

- Displays essential information on the LCD Screen (16x2) for the driver.

- Activates the Buzzer Alert when an anomaly is detected.

- Sends data to the Zigbee Module for wireless transmission.

4. Zigbee Module

This module facilitates wireless communication:

- Transmits the collected sensor data from the ESP32 to the back-end system for further analysis.

5. Back-End System

This system processes and analyzes the transmitted data:

- Machine Learning Model:

- Identifies patterns and predicts potential failures.

- Flags anomalies based on historical and real-time data.

- Database: Stores sensor data, predictions, and historical trends for future analysis and enhancement of the machine learning model.

6. Dashboard (Front-End)

A user interface for monitoring real-time data, alerts, and historical trends.

- It provides actionable insights to the user or maintenance team.

7. Alerts and Notifications

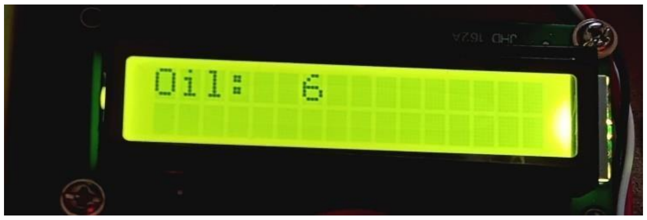
- Buzzer Alert: Provides immediate audio feedback to the driver when critical issues arise.

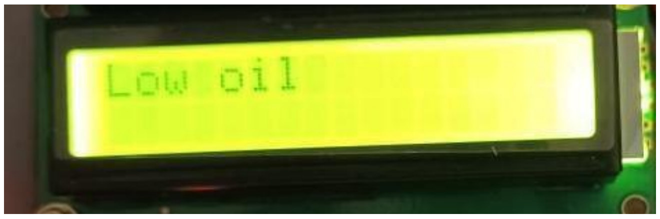
- Telegram Bot: Delivers notifications with detailed information to remote users or maintenance teams, enabling proactive responses.

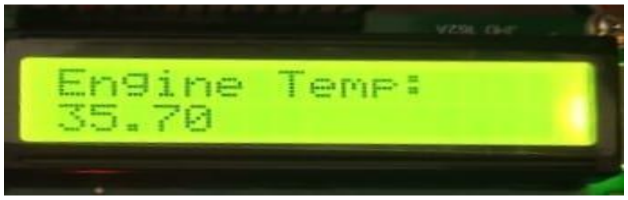
8. Power Supply (PSU)

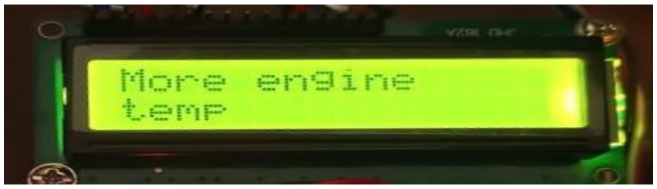
- The system operates on a Lead Acid Battery, guaranteeing dependable functionality for all components.

6.3 Data and alerts mechanism display









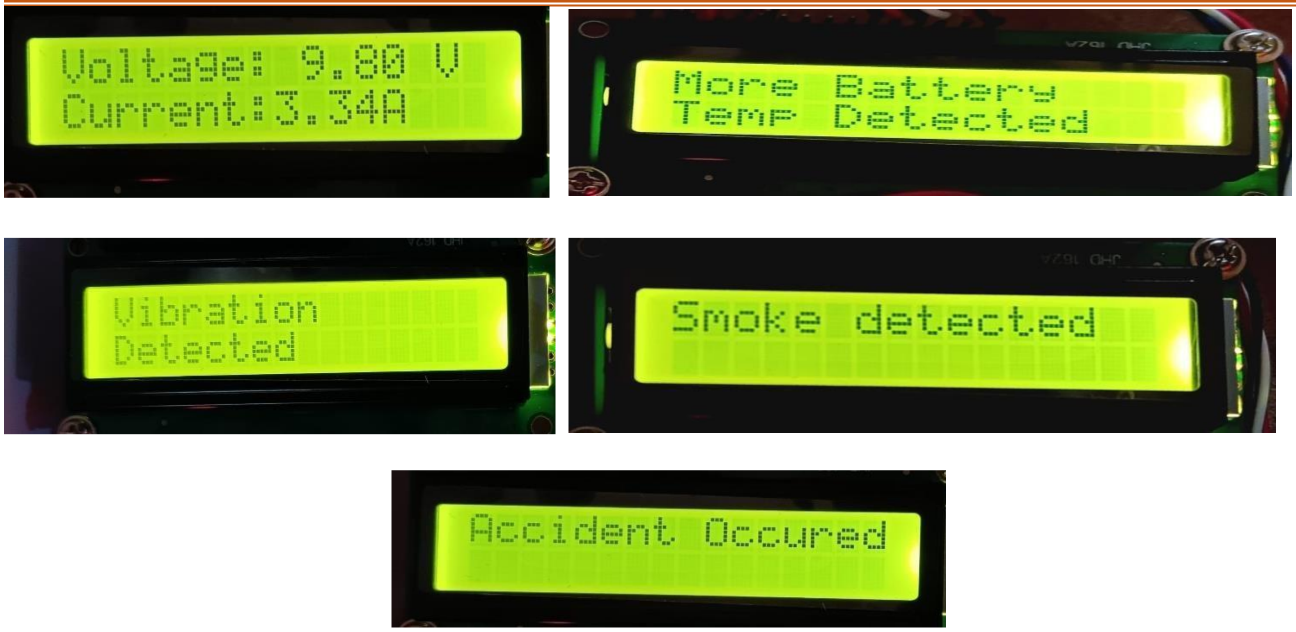


Figure 6.2: Data Display on LCD

6.4 Flow Diagram

The predictive maintenance system starts with its initialization, followed by the collection of data from various sensors, which is then sent to the ESP32 microcontroller. This microcontroller transmits the data to a Flask server using Zigbee for further analysis. The server evaluates the data against set thresholds and employs machine learning models to forecast potential faults or maintenance requirements based on historical data. When an anomaly or problem is identified, the system triggers alerts. These alerts are shown on an LCD, activated by a buzzer, and also sent out remotely through a Telegram bot. The LCD display is monitored to ensure data updates are successful, with retries conducted if needed. Historical data is preserved to enhance the predictive model continuously. This process is repeated for ongoing monitoring, allowing for timely fault detection and reducing the risk of unexpected failures.

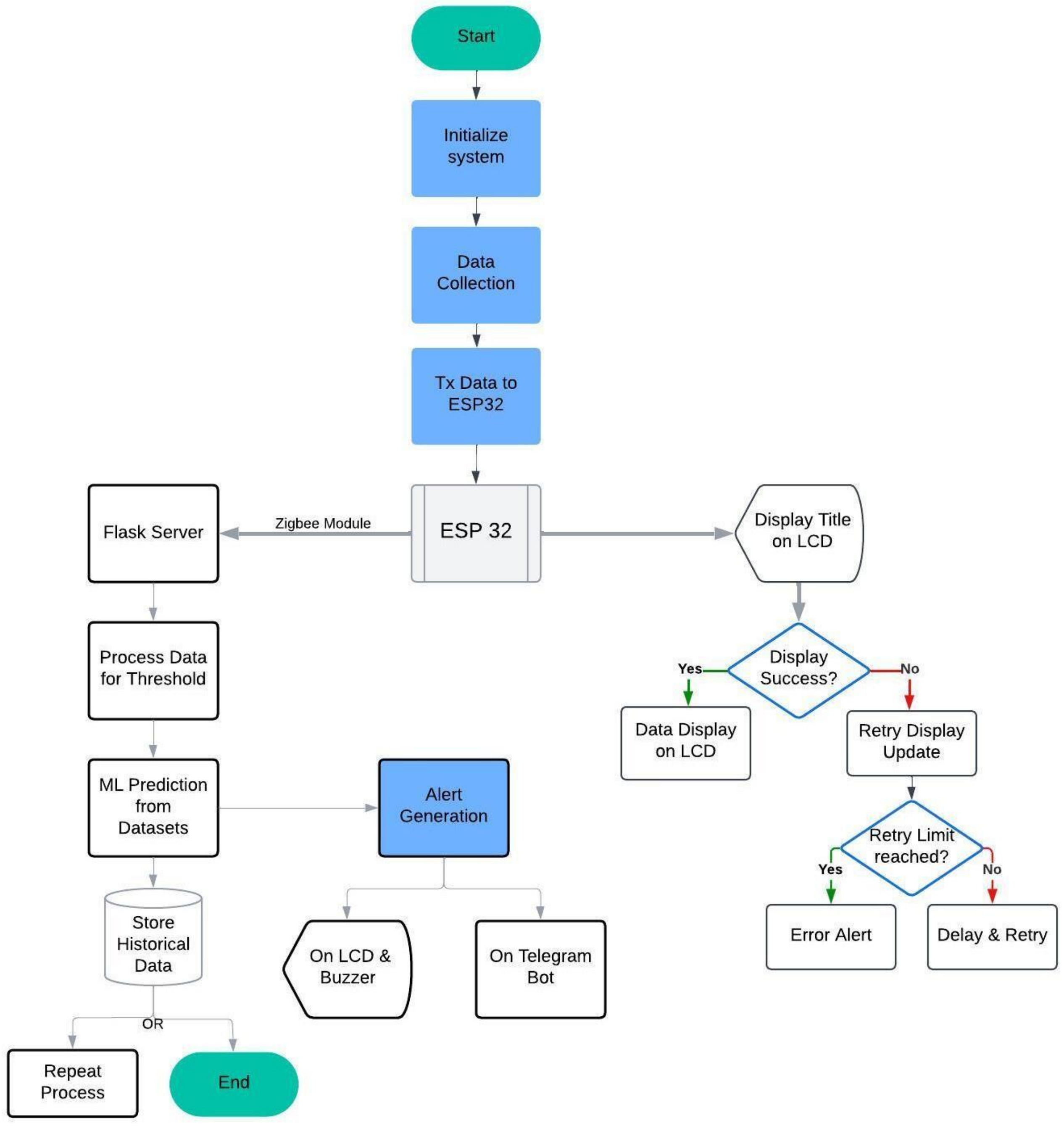


Figure 6.3: Flow Diagram

1. The process starts with the activation of the predictive maintenance system, which can be done either manually or automatically when the vehicle is turned on.

2. Initialize System

- The system sets up all hardware components (like sensors, ESP32, LCD, and Zigbee module) along with software components (including the Flask server and machine learning model).

- Network connections are established to facilitate data communication.

3. Data Collection

- IoT sensors within the vehicle gather real-time data on critical parameters such as:

- Engine temperature

- Vibrations

- Oil pressure

- Tire pressure

- Battery voltage

- This collected data is then sent to the ESP32 microcontroller.

4. Transmit Data to ESP32

- The sensor data is transmitted to the ESP32, which preprocesses and organizes it for further transmission.

5. Display Title on LCD

- The ESP32 shows an initialization or monitoring status message on the vehicle's LCD.

6. Check Display Success

-Success: If the LCD successfully displays the data, the process proceeds to the next step.

-Failure: If the LCD does not display correctly, a retry mechanism is activated.

7. Retry Display Update

- The system attempts to display the message on the LCD again, within a set number of attempts:

- Retry Limit Reached: If the maximum number of retries is reached, an error alert is generated.

- Retry Successful: If the display update is successful, the process continues.

8. Data Display on LCD

- Real-time data collected from the sensors is shown on the vehicle's LCD for the driver to monitor.

9. Communicate with Flask Server

- The ESP32 sends the collected sensor data to the Flask server through the Zigbee module for further processing and analysis.

10. Process Data for Thresholds

- The Flask server processes the incoming data and compares it against predefined thresholds.

- Example:

- If the engine temperature goes beyond a safe limit, it flags a potential overheating issue.

11. Machine Learning Predictions

- The processed data is input into a machine learning model that has been trained on historical datasets. The model forecasts potential failures by analyzing patterns in the data.

12. Generate Alerts

- Based on the analysis and predictions:

- On LCD & Buzzer: Alerts are shown on the LCD screen, and an audible buzzer sounds to immediately notify the driver.

- On Telegram Bot: Alerts are sent to the user or maintenance team through Telegram, offering detailed information about the issue.

13. Store Historical Data

- All processed data and predictions are saved in a database:

- To enhance the accuracy of machine learning models.

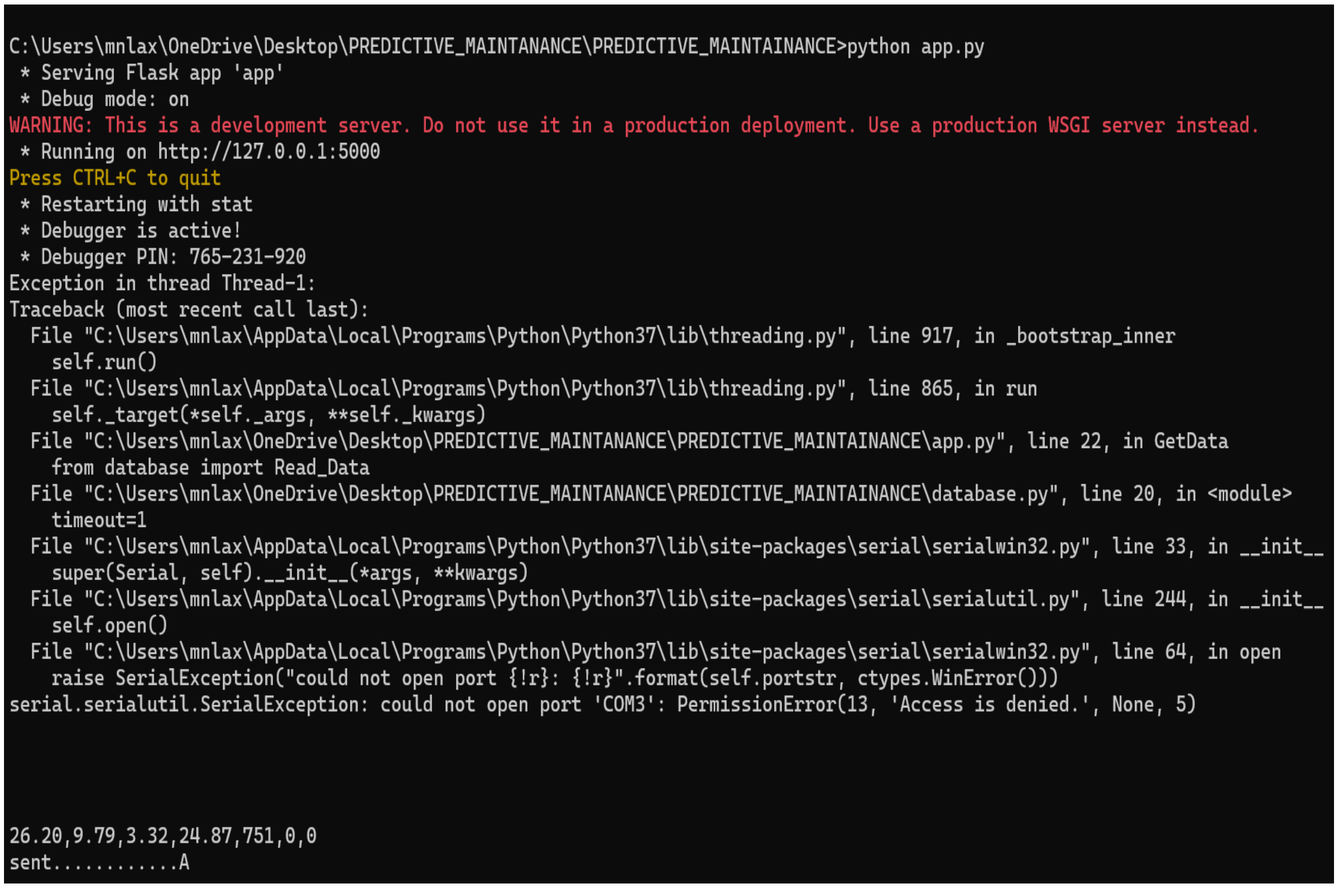
- To conduct long-term trend analysis on vehicle components.

14. Repeat Process or End

- Repeat: If the vehicle is still active, the system continues to loop back to collect and process data.

- End: If the vehicle is turned off or the monitoring session concludes, the process halts.

6.5 Code Execution

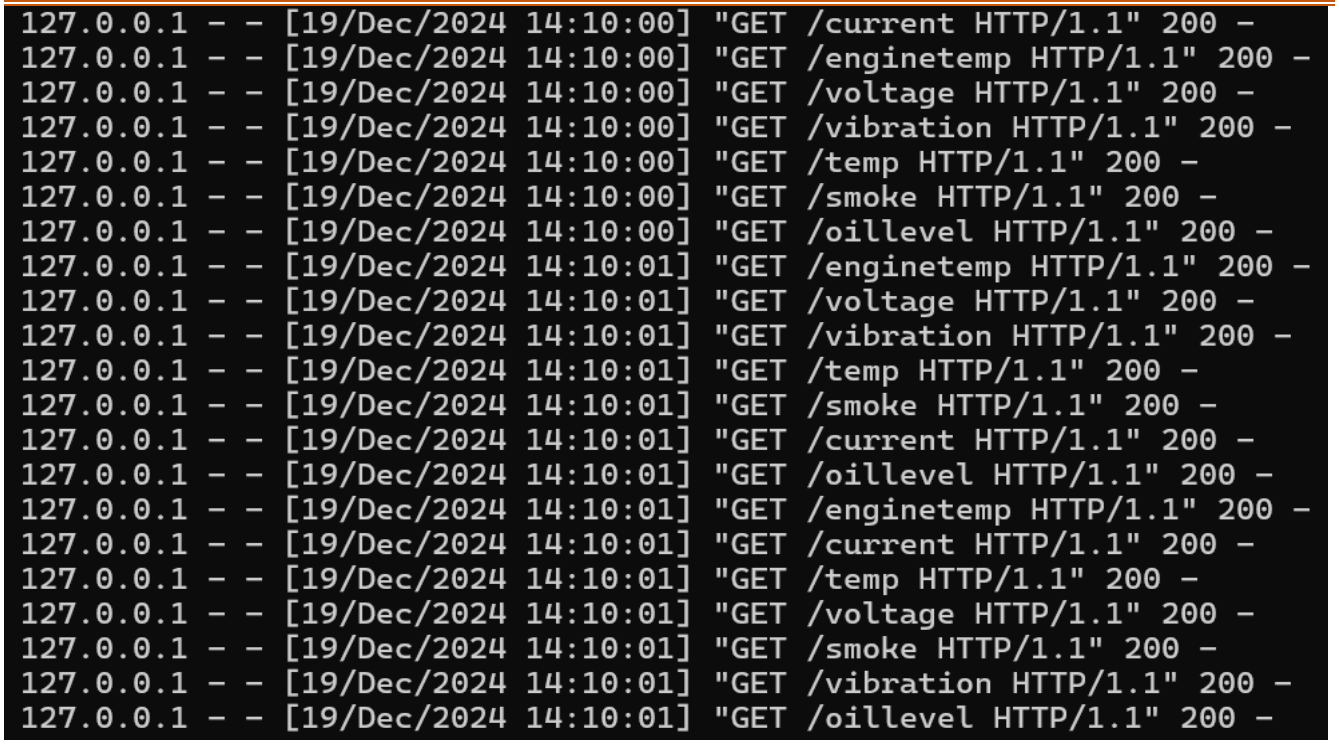


```
C:\Users\mnlax\OneDrive\Desktop\PREDICTIVE_MAINTANANCE\PREDICTIVE_MAINTAINANCE>python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 765-231-920
Exception in thread Thread-1:
Traceback (most recent call last):
  File "C:\Users\mnlax\AppData\Local\Programs\Python\Python37\lib\threading.py", line 917, in _bootstrap_inner
    self.run()
  File "C:\Users\mnlax\AppData\Local\Programs\Python\Python37\lib\threading.py", line 865, in run
    self._target(*self._args, **self._kwargs)
  File "C:\Users\mnlax\OneDrive\Desktop\PREDICTIVE_MAINTANANCE\PREDICTIVE_MAINTAINANCE\app.py", line 22, in GetData
    from database import Read_Data
  File "C:\Users\mnlax\OneDrive\Desktop\PREDICTIVE_MAINTANANCE\PREDICTIVE_MAINTAINANCE\database.py", line 20, in <module>
    timeout=1
  File "C:\Users\mnlax\AppData\Local\Programs\Python\Python37\lib\site-packages\serial\serialwin32.py", line 33, in __init__
    super(Serial, self).__init__(*args, **kwargs)
  File "C:\Users\mnlax\AppData\Local\Programs\Python\Python37\lib\site-packages\serial\serialutil.py", line 244, in __init__
    self.open()
  File "C:\Users\mnlax\AppData\Local\Programs\Python\Python37\lib\site-packages\serial\serialwin32.py", line 64, in open
    raise SerialException("could not open port {!r}: {!r}".format(self.portstr, ctypes.WinError()))
serial.serialutil.SerialException: could not open port 'COM3': PermissionError(13, 'Access is denied.', None, 5)




26.20,9.79,3.32,24.87,751,0,0
sent...........A
```

Figure 6.4: Code Execution 1



```
127.0.0.1 - - [19/Dec/2024 14:10:00] "GET /current HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:00] "GET /enginetemp HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:00] "GET /voltage HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:00] "GET /vibration HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:00] "GET /temp HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:00] "GET /smoke HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:00] "GET /oillevel HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:01] "GET /enginetemp HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:01] "GET /voltage HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:01] "GET /vibration HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:01] "GET /temp HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:01] "GET /smoke HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:01] "GET /current HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:01] "GET /oillevel HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:01] "GET /enginetemp HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:01] "GET /current HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:01] "GET /temp HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:01] "GET /voltage HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:01] "GET /smoke HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:01] "GET /vibration HTTP/1.1" 200 -
127.0.0.1 - - [19/Dec/2024 14:10:01] "GET /oillevel HTTP/1.1" 200 -
```

Figure 6.5: Code Execution 2

6.6 Machine learning model

In the Predictive Maintenance System for automobiles, the Random Forest algorithm is used to forecast potential failures and maintenance requirements based on sensor data. Random Forest is an ensemble learning method that constructs multiple decision trees, each trained on a random subset of the data. The final output is obtained by combining the results from all trees (using mode for classification and average for regression). This algorithm is particularly effective at managing noisy data and complex relationships, which is essential in predictive maintenance, as sensor data can fluctuate due to environmental factors or component wear. It is also well-suited for predicting the Remaining Useful Life (RUL) of components. By utilizing a labeled dataset that includes historical sensor data and corresponding maintenance records, which detail when parts were replaced or failed, the system can make informed predictions.

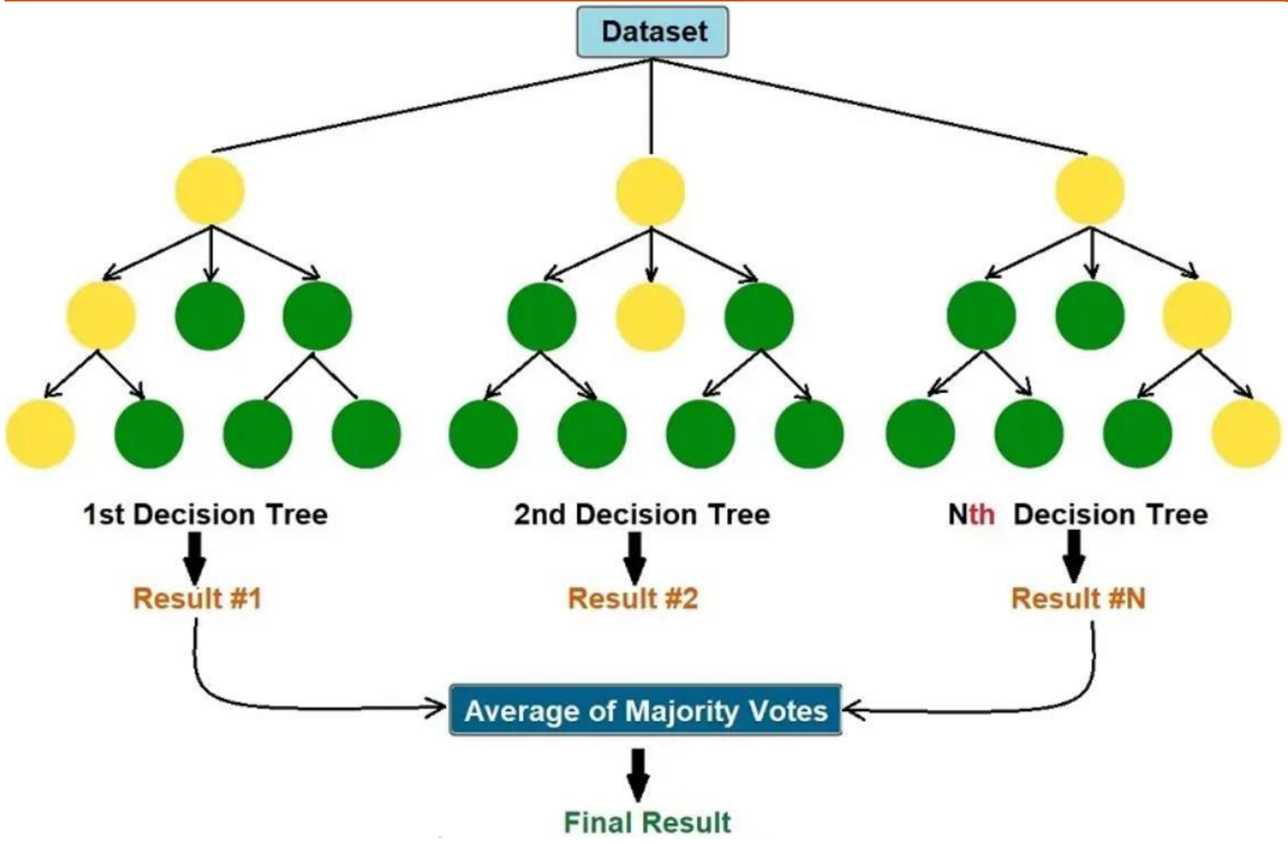
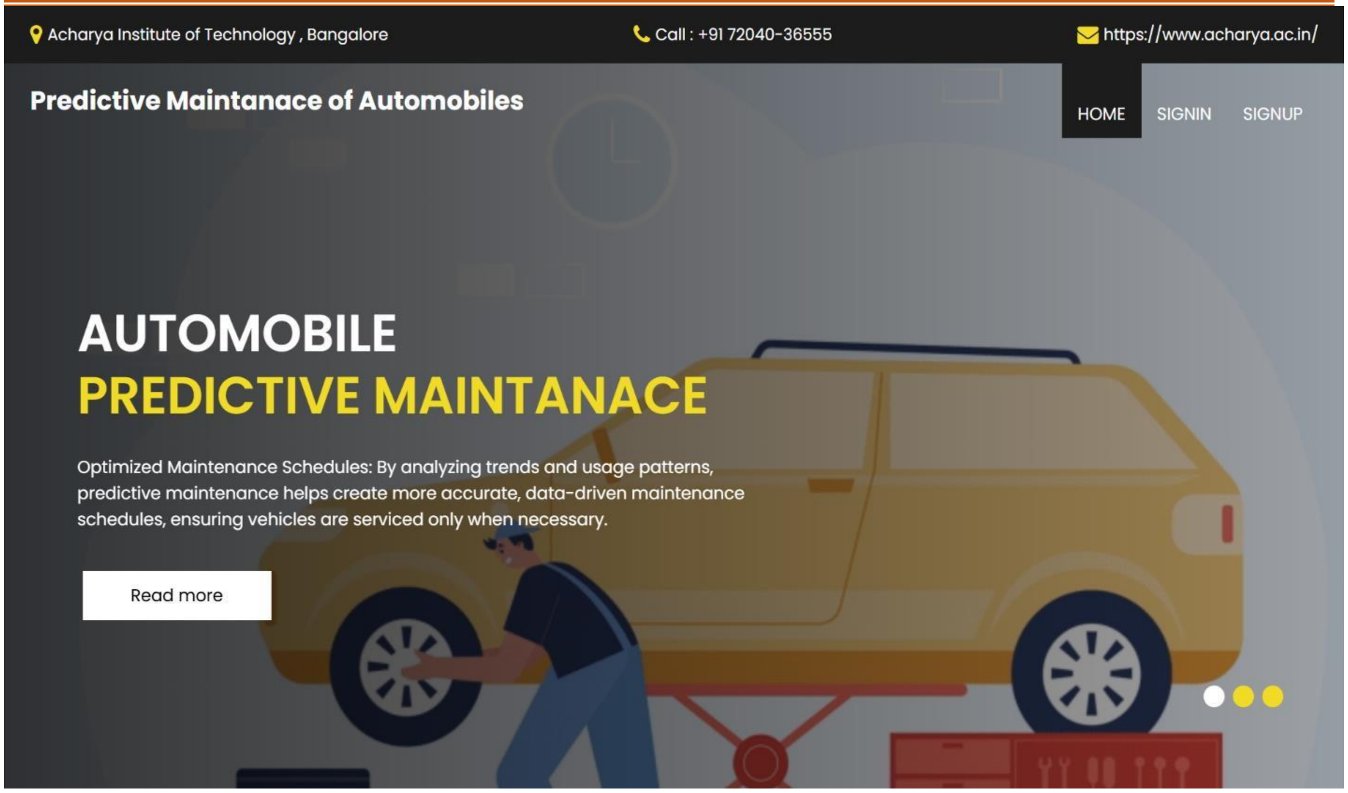


Figure 6.6: Tree Diagram of Machine Learning Model

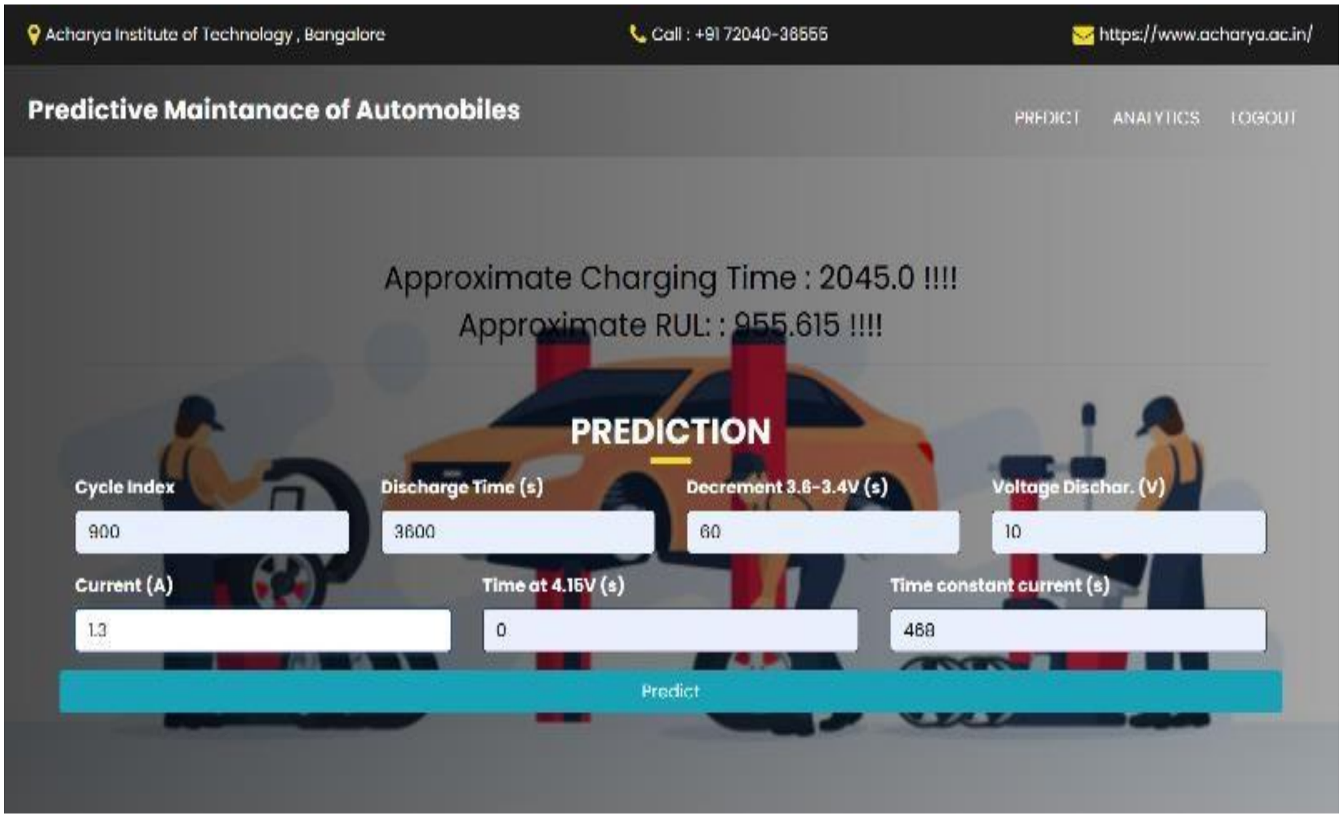
Features:

- Ensemble Learning: This approach combines several decision trees to enhance accuracy and reduce the risk of overfitting.
- Classification and Regression: It is effective for both classifying faults and predicting Remaining Useful Life (RUL).
- Handles Missing Data: The method can work with datasets that have missing values without the need for imputation.
- Feature Importance: It highlights which features play a crucial role in making predictions.
- Parallel Processing: Decision trees can be constructed simultaneously, leading to quicker.
- Scalability: The technique efficiently manages large datasets and supports real-time monitoring.


Acharya Institute of Technology , Bangalore
Call : +91 72040-36555
https://www.acharya.ac.in/
Predictive Maintanace of Automobiles
HOME
SIGNIN
SIGNUP
AUTOMOBILE
PREDICTIVE MAINTANACE
Read more

Optimized Maintenance Schedules: By analyzing trends and usage patterns, predictive maintenance helps create more accurate, data-driven maintenance schedules, ensuring vehicles are serviced only when necessary.

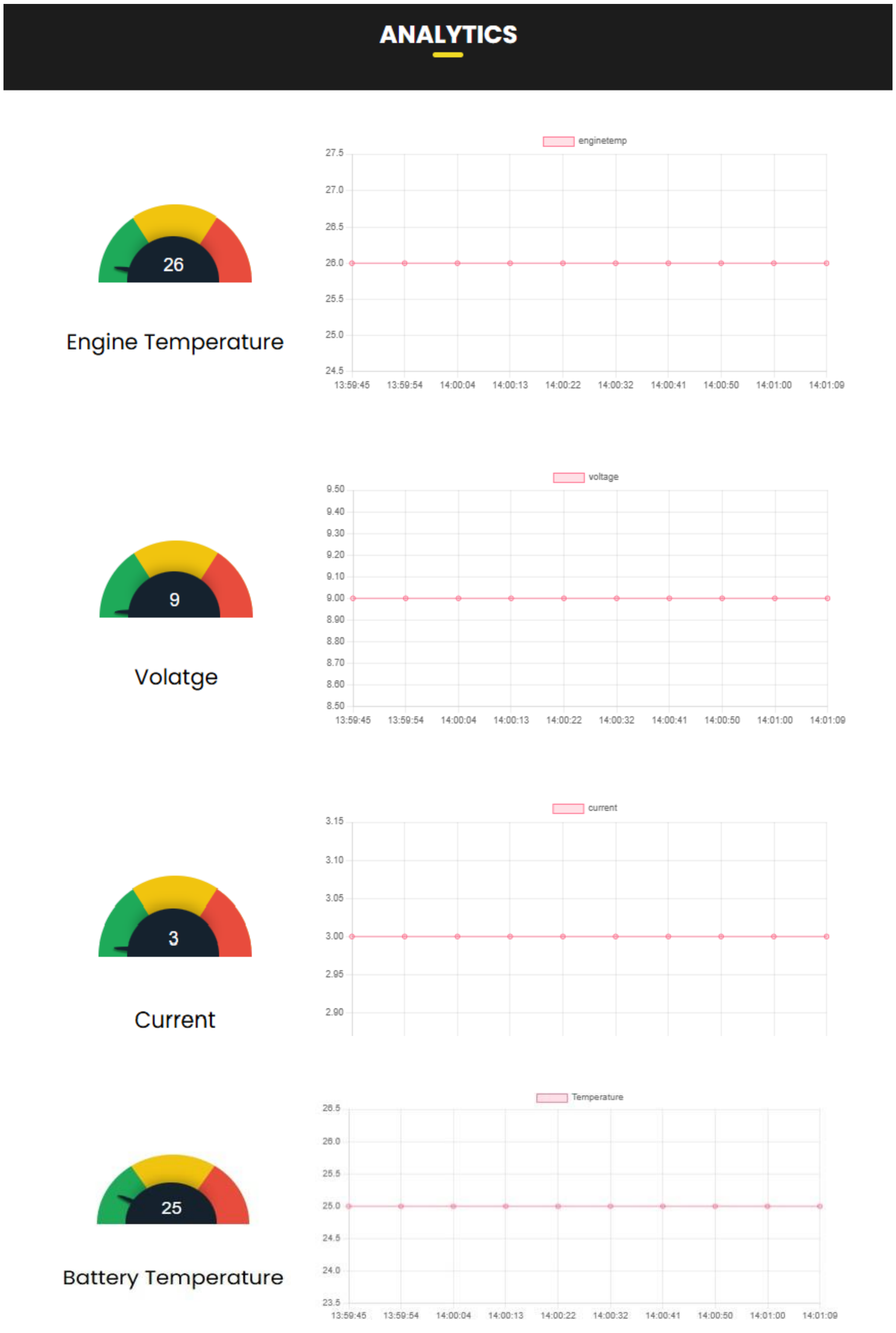
Figure 6.7: Captured web interface 1


Acharya Institute of Technology , Bangalore
Call : +91 72040-36555
https://www.acharya.ac.in/
Predictive Maintanace of Automobiles
PREDICT
ANALYTICS
LOGOUT
PREDICTION
Predict

Approximate Charging Time : 2045.0 !!!!
Approximate RUL: : 955.615 !!!!

Cycle Index: 900
Discharge Time (s): 3600
Decrement 3.6-3.4V (s): 60
Voltage Dischar. (V): 10
Current (A): 1.3
Time at 4.15V (s): 0
Time constant current (s): 468

Figure 6.8: Captured web interface 2


ANALYTICS
26
Engine Temperature
enginetemp
9
Volatge
voltage
3
Current
current
25
Battery Temperature
Temperature
13:59:45
13:59:54
14:00:04
14:00:13
14:00:22
14:00:32
14:00:41
14:00:50
14:01:00
14:01:09

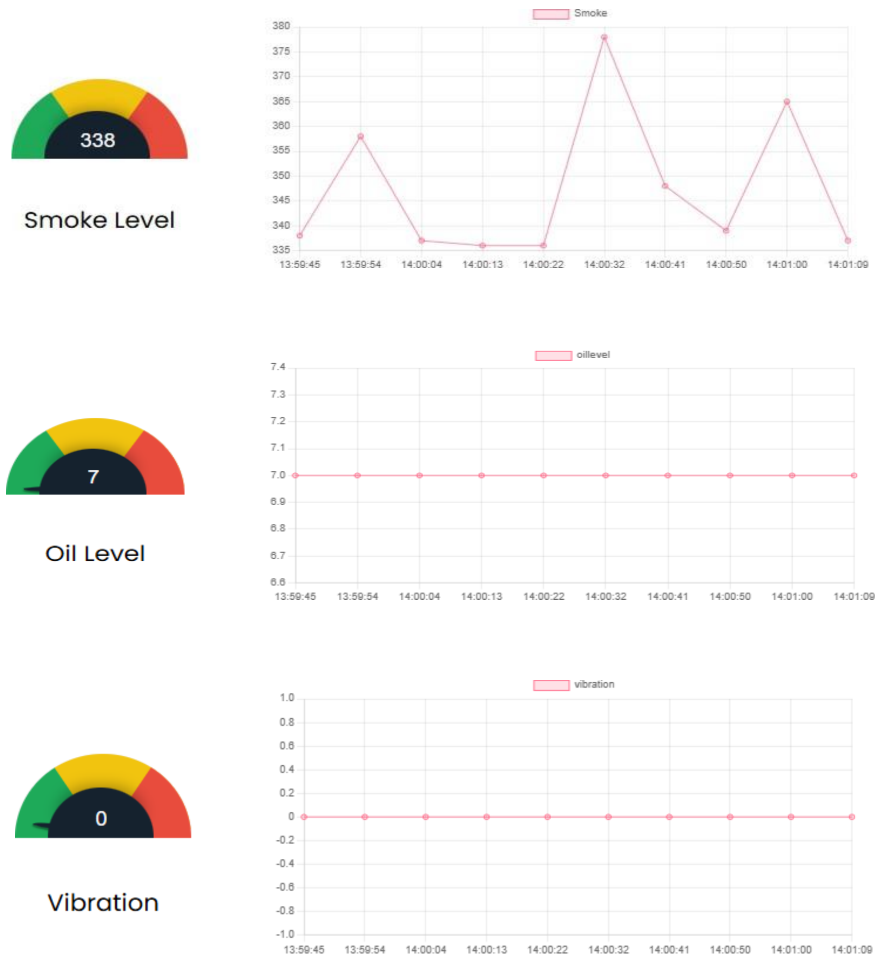


Figure 6.9: Data Analysis of all Sensors

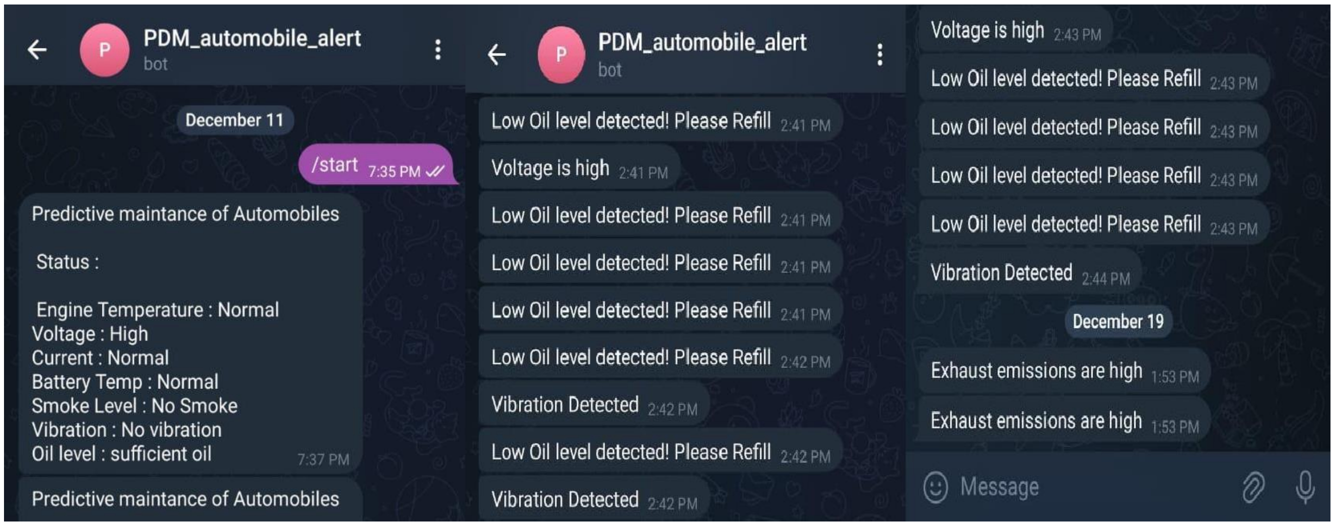
6.5 Telegram bot Notifications

Integrating a Telegram bot into the Predictive Maintenance System for Automobiles starts with the creation and setup of the bot. This is done using Bot Father on the Telegram app, where a new bot is created, and a unique Bot Token is generated to authenticate and interact with the Telegram API. The system gathers real-time data from various IoT sensors, including temperature, vibration, oil level, and motion sensors.

This data is processed by the ESP32 microcontroller, which sends the information to the back-end server using Zigbee or other communication protocols. The back-end server analyzes the incoming sensor data through threshold checks and machine learning models to detect potential faults or anomalies, such as high engine temperature, unusual vibrations, or low oil levels. When an issue is identified, an alert is generated.

The bot also facilitates two-way communication, enabling users to interact with the system. Users can issue commands like /status to get real-time sensor data or /history to access historical trends and logs. These commands are handled by the back-end, which returns the requested data to the user via the bot. Furthermore, for critical alerts, the bot may require users to acknowledge the notification by responding with a specific message.

Real-time updates are maintained through either polling, where the system regularly checks for new alerts, or webhooks, which allow for instant communication between the server and the bot. Once deployed, the bot operates continuously on the server, monitoring sensor data and user interactions. The system can be refined over time by adjusting thresholds and enhancing the accuracy of the machine learning model.


PDM_automobile_alert
bot

December 11

/start 7:35 PM

Predictive maintance of Automobiles

Status :

Engine Temperature : Normal
Voltage : High
Current : Normal
Battery Temp : Normal
Smoke Level : No Smoke
Vibration : No vibration
Oil level : sufficient oil 7:37 PM

Predictive maintance of Automobiles

PDM_automobile_alert
bot

Low Oil level detected! Please Refill 2:41 PM
Voltage is high 2:41 PM
Low Oil level detected! Please Refill 2:41 PM
Low Oil level detected! Please Refill 2:41 PM
Low Oil level detected! Please Refill 2:41 PM
Low Oil level detected! Please Refill 2:42 PM
Vibration Detected 2:42 PM
Low Oil level detected! Please Refill 2:42 PM
Vibration Detected 2:42 PM

Voltage is high 2:43 PM
Low Oil level detected! Please Refill 2:43 PM
Low Oil level detected! Please Refill 2:43 PM
Low Oil level detected! Please Refill 2:43 PM
Low Oil level detected! Please Refill 2:43 PM
Vibration Detected 2:44 PM

December 19

Exhaust emissions are high 1:53 PM
Exhaust emissions are high 1:53 PM

Figure 6.10: Telegram Bot for Data and Alerts

6.6 Advantages and Disadvantages

Advantages:

- Proactive Problem Solving: Identifies issues early on to avoid expensive and inconvenient vehicle breakdowns.

- Economic Benefits: Helps save money by minimizing unnecessary repairs and streamlining maintenance schedules.

- Safety First: Safeguards users by alerting them to critical issues like overheating and potential accidents before they worsen.

- Real-Time Monitoring: Zigbee-powered data transmission provides instant diagnostics, regardless of the vehicle's location.

- Versatile Applications: Functions smoothly for personal cars, commercial fleets, and industrial vehicles alike.

- Eco-Friendly Approach: Reduces waste by extending component lifespan and decreasing the need for premature replacements.

Disadvantages :

Upfront Investment: The high initial costs associated with sensors and infrastructure might discourage potential users.

Connectivity Challenges: A reliable network connection is necessary, which may not be accessible in certain regions.

Technical Expertise Needed: The installation and maintenance processes require specialized knowledge in IoT and machine learning.

Component Maintenance: Sensors and systems need regular inspections, contributing to ongoing maintenance efforts.

CHAPTER: 07

FUTURE ENHANCEMENT

1. Smarter Models with Deep Learning Utilize LSTM networks to effectively analyze time-series data, enhancing the accuracy of anomaly predictions. Employ CNNs for recognizing spatial patterns, which improves the classification of faults in sensor readings. Incorporate adaptive learning features that enable the system to refine its predictions in real-time as it gathers more data.

2. Cloud Intelligence for Scalability Shift data processing and visualization to cloud services like AWS IoT Core or Google Cloud IoT, increasing computational capabilities and scalability. Offer real-time global insights through sophisticated dashboards that can be accessed from any device.

3. Expanding the Sensor Ecosystem Install sensors in vehicles to monitor tire pressure, fuel efficiency, and emissions, turning the system into a comprehensive maintenance solution. Choose weather-resistant and durable sensors to guarantee dependable performance in harsh weather conditions.

4. Mobile-First User Experience Create a user-friendly mobile app that enhances Telegram notifications with in-depth diagnostics, predictive reports, and service reminders. Allow for remote control and real-time updates, giving users the ability to manage their vehicle's health right from their devices.

5. Predictive Maintenance as a Subscription Transform the solution into a Predictive Maintenance as a Service (PMaaS) model, providing customized plans for both individual users and fleet operators. Incorporate premium features such as personalized analytics, and automated maintenance scheduling.

6. Fortified Data Privacy and Security Adopt strong encryption methods to protect user data during both transmission and storage. Ensure compliance with regulations like GDPR to build user trust and maintain data integrity.

CHAPTER : 08

RESULTS

1. Seamless Real-Time Monitoring: Effectively tracked essential parameters such as temperature, vibrations, battery health, oil levels, and gas presence. Set up accurate anomaly thresholds for quick detection of potential problems.

2. Intelligent Predictive Maintenance: Utilized the Random Forest Algorithm to reliably predict maintenance requirements based on both real-time and historical data. Proactively notified users about risks like overheating or low battery levels with high accuracy. Estimated the Remaining Useful Life (RUL) of components, helping users schedule maintenance efficiently.

3. Interactive Notification System: A Telegram bot provided immediate delivery of detailed alerts regarding faults or anomalies. Allowed users to interact with the system by checking real-time statuses and analyzing historical performance trends.

4. Reliable System Architecture: The ESP32 microcontroller efficiently handled data flow from various sensors while ensuring seamless transmission through Zigbee. A user-friendly LCD interface allowed drivers to quickly access essential vehicle metrics.

5. Enhanced Safety Protocols: It identified critical situations such as excessive vibrations and engine overheating, which helped reduce the chances of breakdowns. Vibration sensors effectively activated accident detection alerts, facilitating prompt responses during emergencies.

6. Optimized Costs and Efficiency: There were notable decreases in unexpected repairs and downtime, resulting in lower operational costs for users. Enhanced vehicle performance and reliability reduced disruptions, improving efficiency for both fleets and individual vehicles.

7. Robust Validation and Testing Machine learning models: underwent thorough testing on a variety of datasets, achieving high accuracy in fault and anomaly predictions. Sensor readings were validated against actual maintenance events, confirming the system's reliability in real-world applications.

8. Comprehensive Dashboard Insights: An intuitive dashboard was provided, displaying real-time sensor data, actionable alerts, and historical analysis. This improved decision-making by visualizing vehicle health trends and offering a clear overview of maintenance needs.

REFERENCES

[1] Kanawaday, Ameeth, and Aditya Sane. "Machine learning for predictive maintenance of industrial machines using IoT sensor data.", In 2017 8th IEEE international conference on software engineering and service science (ICSESS), pp. 87-90. IEEE, 2017..

[2] yvaz, Serkan, and Koray Alpay. "Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time.", Expert Systems with Applications 173 (2021): 114598..

[3] Cakir, Mustafa, Mehmet Ali Guvenc, and Selcuk Mistikoglu. "The experimental application of popular machine learning algorithms on predictive maintenance and the design of IIoT based condition monitoring system.", Computers & Industrial Engineering 151 (2021): 106948..

[4] Kalusivalingam, A. K., Sharma, A., Patel, N., & Singh, V. (2020). Enhancing Predictive Maintenance in Manufacturing Using Machine Learning Algorithms and IoT-Driven Data Analytics., International Journal of AI and ML, 1(3)..

[5] C. V. S. M. D. S. Ravi Aravind, Machine Learning Applications in Predictive Maintenance for vehicles, International Journal Of Engineering And Computer Science, 11 November 2022.

[6] Predictive Maintenance – Bridging Artificial, 2021, Gerasimos G. Samatas, Seraphim S. Moumgiakmas, George A. Papakostas*.

[7] Chen, C., Liu, Y., Wang, S., Sun, X., Di Cairano-Gilfedder, C., Titmus, S. and Syntetos,, Advanced Engineering Informatics, 44, p.101054. 10.1016/j.aei.2020.101054, A.A., 2020. Predictive maintenance using cox proportional hazard deep learning..

[8] Y. L. Y. C. C. a. J. Z. Liang, Extracting topic-sensitive content from textual documents—A hybrid topic model approach. Engineering Applications of Artificial Intelligence,, 2018.

[9] C. L. Y. S. X. C.-G. D. T. S. Chen, Automobile maintenance modelling using gcForest., IEEE 16th International Conference on Automation Science and Engineering (CASE) 10.1109/CASE48305.2020.9216745, In 2020.

[10] M. A. M. D. A. V. P. Nithish Kanna J L Krishnakumar G, Predictive Maintenance of Motors using Machine Learning, Kumaraguru College, Issue 4 April 2024.

DrillBit

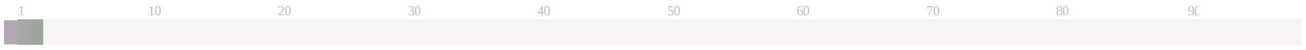
The Report is Generated by DrillBit Plagiarism Detection Software

Submission Information

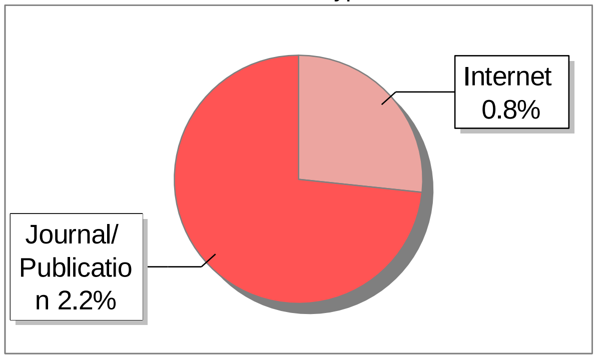
Author Name	Laxman M Nayanegali
Title	Project Report
Paper/Submission ID	2934543
Submitted by	asklibrarian@acharya.ac.in
Submission Date	2025-01-03 11:43:56
Total Pages, Total Words	44, 7350
Document type	Project Work

Result Information

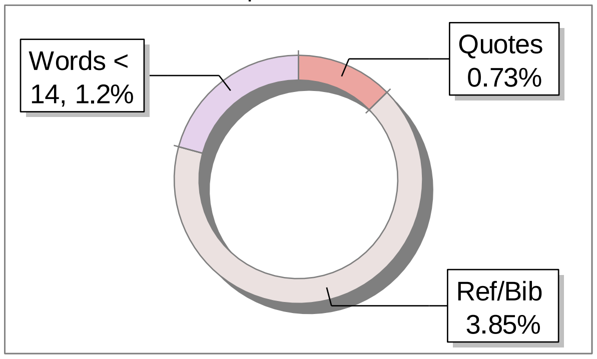
Similarity **3 %**



Sources Type



Report Content



Exclude Information

Quotes	Excluded
References/Bibliography	Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

3	**12**	**A**	A-Satisfactory (0-10%)
SIMILARITY %	MATCHED SOURCES	GRADE	B-Upgrade (11-40%)
			C-Poor (41-60%)
			D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	orca.cardiff.ac.uk	<1	Publication
2	www.e3s-conferences.org	<1	Publication
3	www.ijcrt.org	<1	Publication
6	ebin.pub	<1	Internet Data
7	The next evolution of MDE a seamless integration of machine learning by Hartmann-2017	<1	Publication
8	Learning Personalized Discretionary Lane-Change Initiation for Fully Autonomous by Liu-2020	<1	Publication
9	www.eelet.org.uk	<1	Publication
10	springeropen.com	<1	Internet Data
11	springeropen.com	<1	Internet Data
12	www.freepatentsonline.com	<1	Internet Data
13	moam.info	<1	Internet Data
14	www.dx.doi.org	<1	Publication