An Intelligent IoT Architecture for Continuous Vehicle Health Monitoring and Early Fault Detection

Gundu Naga Sowmya[1], Divi Tejaswini[2],Manitha PV [3]
Department of Electrical and Electronics Engineering
Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham, India
bl.en.p2ebs25007@bl.students.amrita.edu[1], bl.en.p2ebs25014@bl.students.amrita.edu[2],
pv_manitha@blr.amrita.edu[3]

***Abstract*—Monitoring of key parameters of modern vehicles is essential for their safe operation, reliable functionality, and optimal performance. Unusual situations like overheating, high power consumption, and motor overload may cause equipment failure, inefficiency, and other faults. Conventional methods of monitoring do not include real-time monitoring, remote accessibility, and fault predictions.**

The present paper considers a Smart Vehicle Health Monitoring System based on the IoT and ML technologies. The system is designed on the basis of the ESP32 microcontroller, LM35 temperature sensor, ACS712 current sensor, BO motor, alerting buzzer, and Blynk cloud server for monitoring and fault detection. Data from sensors are collected in real-time mode by ESP32 microcontroller and analyzed in terms of classification of vehicle operation as NORMAL, WARNING, and CRITICAL modes. In order to increase the predictive maintenance potential, several machine learning techniques like Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, and K-Nearest Neighbor have been used by employing the acquired data set. According to experimental results, the Random Forest technique proved to be the most accurate classification technique, which was 87.3

The processed data as well as system status can be transferred to the Blynk cloud platform via WiFi. The buzzer alerting system can be triggered under the abnormal operating conditions like overheating and overload of the vehicle. Thus, the developed system provides a low cost and intelligent solution for real-time diagnostics and maintenance of the vehicles.

***Index Terms*—IoT, ESP32, Vehicle Health Monitoring, LM35 Sensor, ACS712 Current Sensor, Blynk Cloud, Embedded Systems, Real-Time Monitoring, Fault Detection, Predictive Maintenance, Smart Vehicles, Cloud Monitoring, Edge Computing, Automotive Diagnostics, Wireless Sensor Monitoring.**

I. Introduction

Present-day automobiles have developed into advanced embedded systems, which must be monitored to guarantee their safety, reliability, and effective operation. Such factors as temperature, current, voltage, and loads play an essential role in automotive performance. In case of overheating and overload, motors become damaged, suffer from electrical malfunctions, lose efficiency, and break down unexpectedly. Conventional monitoring systems are generally performed manually and do not provide timely diagnoses, remote access, or preventive maintenance. Recent researches on IoT technologies allow for effective monitoring and fault detection in automotive systems [1]-[4].

Modern automobiles include a variety of electronic subsystems that are responsible for ensuring their safe and effective operation. All of these subsystems interact constantly in order to achieve this goal through their embedded controllers and communication systems. Advanced technologies such as sensor installation and wireless communication have facilitated the diagnostic process and vehicle automation significantly. With growing system complexity, however, it becomes increasingly challenging to monitor the condition of the automobile and detect any malfunctions. Consequently, reliable and affordable monitoring is extremely relevant to modern automotive systems [5]-[6]. The advancements in the field of embedded systems and IoT technologies have transformed the vehicles into a cyber physical system. They can communicate and take intelligent decisions remotely. Scientists have conducted studies on integration of sensors, wireless modules, cloud computing and edge computing for diagnostics and predictive maintenance of vehicles [7]-[9].

The scientists Rajesh Kumar et al. developed an IoT-based vehicle monitoring system with the help of temperature and vibration sensors to achieve vehicle diagnostics and predictive maintenance [10]. It proved through their study that the monitoring via cloud makes the vehicle safe as well as efficient in its functioning. On the other hand, Patel et al. presented a wireless health monitoring system for automobiles by utilizing embedded sensors along with wireless technology [11]. Sharma et al. developed a predictive maintenance strategy for electric cars employing machine learning and cloud computing [12]. This was achieved through the use of sensor data for detecting abnormalities and predicting potential failure prior to any failure occurrence. In a similar case, Ramesh et al. developed an IoT based monitoring system for the monitoring of motors employing temperature and current sensors [13].

The monitoring was done to detect overload situations that might lead to motor failure.

In recent IEEE journal articles focusing on smart transportation systems, researchers have investigated the role of embedding sensors, edge computing, cloud dashboards, and wireless communication in vehicle diagnostics and predictive maintenance [14]-[15]. However, despite such improvements brought by existing systems, there are problems that need to be addressed, including costly implementation, inadequate edge computing capabilities, inability to classify faults in real time, and unavailability of affordable cloud-based visualization tools. For this reason, this paper suggests a Smart Vehicle Health Monitoring System consisting of ESP32, LM35 Temperature Sensor, ACS712 Current Sensor, BO Motor, Buzzer Alarm, and Blynk Cloud Dashboard.

This solution monitors temperature and current values in real time, detects possible faults on the edge, identifies operational statuses of either NORMAL, WARNING, and CRITICAL types, and communicates data to a cloud dashboard through WiFi communication protocol.

II. Proposed System

The suggested Smart Vehicle Health Monitoring System aims at monitoring critical parameters of the vehicle such as temperature and current through the Internet of Things technology. The suggested system makes use of ESP32 as the processing and communication module, LM35 sensor to measure the temperature, ACS712 sensor to measure the motor current, BO motor as the dynamic load, buzzer as the fault notification module, and Blynk cloud-based platform to provide real-time monitoring.

The LM35 sensor measures the temperature readings of the motor and electric component. On the other hand, the ACS712 sensor measures the motor current under varying loads. After the acquisition of the sensor values, they are further processed by the ESP32 microcontroller where edge-level processing takes place. Based on the acquired threshold conditions, the system determines its operational state.

According to the obtained sensor values, the suggested system categorizes the working state of the vehicle into:

NORMAL WARNING CRITICAL Under normal operation, the system keeps updating the sensor readings to the cloud dashboard by utilizing WiFi for communication. In situations where abnormal operation is encountered due to overheating or overload, the ESP32 triggers the buzzer alarm and updates the system status to the cloud dashboard.

The design for the system entails the use of multi-layer IoT architecture, which comprises the sensing layer, edge processing layer, communication layer, cloud layer, and application layer. Under the multi-layer IoT architecture, the sensing layer is used to collect data from the sensors, edge processing layer is utilized for analyzing data at the edge through ESP32, while the communication layer uses WiFi for transferring data. The proposed system has many advantages, such as:

Vehicle tracking in real-time Visualization through cloud Affordable cost of setup Fault detection at the edge Proactive maintenance system Scalable Internet of Things architecture Web dashboard remote monitoring

In contrast with the existing vehicle tracking systems, the proposed vehicle tracking system conducts local fault analysis, then sends the results to the cloud, which helps decrease latency and increase system reliability. Temperature and current sensors provide more accurate fault detection and improve vehicle safety.

As shown in the block diagram of the proposed system, LM35 temperature sensor and ACS712 current sensor are connected to the ESP32 microcontroller. The ESP32 interacts with the Blynk cloud server through WiFi communications.

*1) **Machine Learning Integration**:* Besides the real-time monitoring, machine learning algorithms will be used in order to predict the health of the vehicle. The temperature and current values obtained through the real-time monitoring system are stored and used as input to train classification algorithms. The dataset is classified into three categories: NORMAL, WARNING, and CRITICAL. Different machine learning algorithms such as Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, and K-Nearest Neighbor are tested and the predictions are made based on sensor readings.

III. METHODOLOGY

The designed Smart Vehicle Health Monitoring System follows a comprehensive approach that includes sensing, edge computing, cloud communication, live monitoring, and fault diagnosis based on Internet of Things (IoT) architecture. This process includes various stages to facilitate real-time monitoring and effective vehicle diagnosis.

*1) **Data Acquisition**:* In the initial stage, live data from sensors is obtained using the LM35 temperature sensor and ACS712 current sensor. The LM35 sensor provides real-time temperature readings of the motor and electronics, while the ACS712 sensor determines the current consumed by the motor under various load conditions.

*2) **Sensor Interfacing**:* Both of these sensors interface with the ESP32 microcontroller through the analog input port. The ESP32 microcontroller continuously captures analog input data from sensors and converts them to digital values.

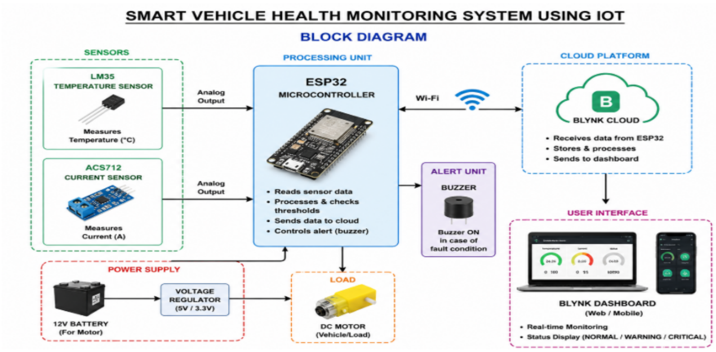

Fig. 1. Architecture of the proposed IoT-enabled smart vehicle health monitoring system

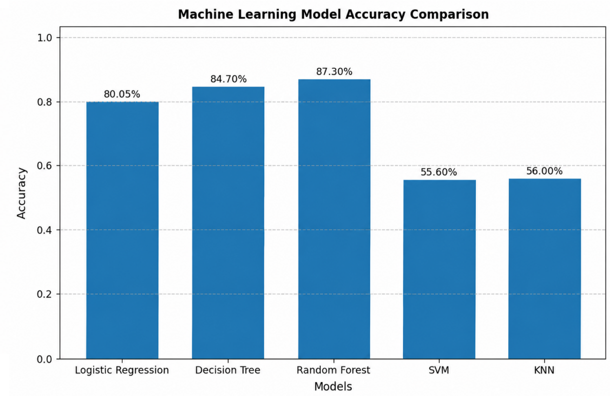

Fig. 2. Machine Learning Model Comparision Dashboard

*3) **Data Processing**:* The processing of the collected data involves converting them into the relevant parameters such as the temperature in degree Celsius and current in amperes.

*4) **Threshold-Based Fault Detection**:* Comparison is made between the processed sensor data and predefined threshold values for abnormal situations. Operating situation is categorized as follows:

NORMAL WARNING CRITICAL

In case of excessive value in temperature or current, then the situation will be identified as faulty situation.

*5) **Edge Level Analysis**:* Analysis and decisions are made by the ESP32 locally before transferring the data to the cloud platform. Edge level analysis cuts down communication delays and enhances response during critical faults.

*6) **IoT Platform**:* WiFi communication is used between ESP32 and the Blynk platform. Virtual pins are used for transferring sensor data like temperature and current data of the system to the cloud platform.

*7) **Cloud Monitoring and Visualization**:* The Blynk cloud platform monitors and visualizes sensor data and system status at a web dashboard. Vehicle health status monitoring is done by a remote user using the web dashboard.

*8) **Alert Mechanism**:* A buzzer alarm system is embedded to provide immediate alerts when fault occurs. When there is a case of an overload condition, the buzzer gets activated.

*9) **Machine Learning Based Classification**:* The dataset that has been collected from sensors is preprocessed for training various machine learning algorithms. Features like temperature and current readings are considered as input parameters for the algorithm. Classification of vehicles' running condition into three different categories – NORMAL, WARNING, and CRITICAL – is performed by the machine learning algorithms.

A. SOFTWARE IMPLEMENTATION

The proposed Smart Vehicle Health Monitoring System software design is implemented using the Arduino Integrated Development Environment (IDE) with the programming language Embedded C/C++. The ESP32 microcontroller is used in this design as the main hardware and communications processor of the system. The software performs various tasks including sensor data collection, data processing, fault detection, communication with the cloud and monitoring via the Blynk interface.

The first step in software design is including the necessary libraries like WiFi and Blynk libraries to enable wireless communication and connectivity with the cloud. Then the ESP32 is programmed with WiFi credentials and Blynk credentials. When the device successfully initializes, it connects to the cloud server using WiFi communication.

For the sensor inputs, the ESP32 communicates with the LM35 temperature sensor and ACS712 current sensor. These two sensors are interfaced to analog input channels of the ESP32. In software, continuous reading of analog values of sensors is performed by the ESP32's built-in ADC function. The analog input values read are converted to voltage values and then to temperature and current values. The value of the temperature is calculated from the voltage reading obtained from the LM35 temperature sensor, which is directly proportional to temperature in degrees centigrade. Likewise, the value of the current is obtained from the current sensor sensitivity and offset voltage readings of the ACS712 current sensor. Techniques such as noise reduction and averaging are used in the software to increase sensor precision and stability in readings.

The fault detection algorithm uses thresholds to classify abnormal operating conditions in the ESP32 program code. The software code continuously checks whether the temperature and current values exceed or fall within their corresponding thresholds. The result of this check enables the software to classify an operating state as NORMAL, WARNING, and CRITICAL.

During normal operation, the ESP32 sends readings from the temperature and current sensors to the cloud server via the virtual pin interface at intervals. The Blynk application monitors the status of the sensor by displaying temperature, current, and status readings in its dashboard. When sensor readings surpass their threshold levels, the alarm buzzer is activated, and the dashboard updates the operating status. Similarly, the software implementation also implements edge-based processing whereby the information collected by the sensors is processed at the edge level by the ESP32 prior to uploading it to the cloud platform. This has the advantage of reducing delays and improving reliability.

In conclusion, the software implementation can efficiently monitor, detect faults, connect with the cloud platform, and be remotely accessed.

1) Implementation Challenges: There were some difficulties associated with the implementation of the recommended Smart Vehicle Health Monitoring System that included difficulties related to the interface design, data collection, cloud connectivity, and fault detection in real time.

The difficulty in measuring the accurate reading through the ACS712 current sensor was one such challenge since the readings of the current had fluctuating and electrical noise due to which there were unstable current readings. The way to solve this problem was through averaging and filtering of the data collected through ESP32 software. Network interruptions sometimes caused a delay in updating information on the dashboard. However, this problem was solved using reconnection mechanisms in the software code.

Threshold calibration for detection of faults was another significant challenge during implementation. Appropriate threshold levels for temperature and current values had to be determined to enable classification of operating conditions into NORMAL, WARNING, and CRITICAL categories. Several experimental trials were carried out using varying loads to determine proper threshold levels.

Proper power management and grounding were yet other crucial issues faced during implementation. As the motor load was powered separately from the ESP32, initial power levels resulted in unstable measurements because of poor grounding. A common ground between all components helped resolve the problem. The real-time processing capability and fault analysis at the edge were critical aspects that needed efficient coding in ESP32 microcontroller. The software had to constantly collect values from the sensor, process the data, communicate with the cloud platform, and trigger alarm systems without any lags.

One more problem was related to incorporating different hardware components into one embedded system, which included the sensor, motor, buzzer, WiFi connectivity, and cloud dashboard. Synchronization of hardware and software modules was crucial for effective system operation.

However, despite all the mentioned difficulties, it was possible to implement the suggested design and prove its efficiency and usefulness for the real-time monitoring of faults in smart vehicle health monitoring applications.

B. Hardware Implementation

Hardware components that will be used in implementing the Smart Vehicle Health Monitoring System include ESP32 microcontroller, LM35 temperature sensor, ACS712 current sensor, BO motor, buzzer alert system, power supply, and Blynk cloud integration. The hardware configuration will be set to continuously monitor the temperature and current parameters and detect faults in real time by using IoT technology.

The ESP32 microcontroller is responsible for performing all processes that include receiving signals from sensors, analyzing the signal data, detecting thresholds and faults, and sending the result to the cloud through WiFi communication. The ESP32 was chosen due to the presence of an integrated WiFi module, low power consumption, and fast processing speed.

An LM35 temperature sensor will be used to detect the temperature of the motor and electrical parts of the vehicle. LM35 outputs an analog voltage level that is equal to the value of temperature detected in degree Celsius. The output of the temperature sensor is connected to the input GPIO35 of the ESP32 board. To calculate the current taken by the motor at various operating conditions, an ACS712 current sensor is employed. An ACS712 sensor uses the Hall effect method of current sensing, which generates an analog output based on the magnitude of the current. The output pin from the ACS712 current sensor is connected to GPIO34 of the ESP32 board. Current flowing from the motor is measured using the ACS712 module to check for any abnormal operation conditions or overloads.

A BO motor is used as a dynamic load to simulate the various conditions under which a vehicle runs. The load provided to the motor during experiments is changed manually. The motor is wired via the ACS712 sensor to enable current measurement.

Buzzer fault indication is added to the system. In case of any fault like overheating or overload, the ESP32 board triggers the buzzer to warn the user of any abnormalities in the system. The buzzer is connected to GPIO2 of the ESP32. Additionally, there is a provision for the power supply for the ESP32 and motor control hardware in the design. A proper grounding system is put in place for the ESP32, sensor components, and motor powering to enhance stability in the process of data acquisition.

ESP32 is able to interface with the Blynk app via wireless WiFi technology. Current values such as temperature, current levels, and the system status are transferred to the cloud dashboard via the network interface.

The overall hardware configuration enables low-cost and efficient real-time monitoring ideal for smart vehicle diagnostics and predictive maintenance operations.

1) Overview of the System: The suggested Smart Vehicle Health Monitoring System is a smart IoT monitoring platform that aims to ensure continuous monitoring and detection of faults in the vehicles in real time. This IoT platform uses two sensors named LM35 and ACS712 for monitoring vehicle temperature and current respectively by interfacing with the ESP32 microcontroller board.

The main aim of designing the suggested IoT-based system is to enhance the safety of the vehicles by detecting abnormal conditions like overheating and overloading of motors in the early stages itself. The ESP32 microcontroller gathers the data from the sensors and analyzes the collected data at the edge level and classifies the overall condition as normal, warning, or critical depending upon pre-defined threshold levels.

This analyzed data from sensors is sent over WiFi to the Blynk platform which provides remote real-time monitoring and visualization of the system parameters. In case there is any abnormality like overheating and excess current, ESP32 triggers the alarm system to notify the user and changes the status on the dashboard. Therefore, it ensures timely identification of the fault and helps maintain the car system proactively.

The suggested system involves a hierarchical Internet of Things structure that includes a sensing layer, edge computing layer, communication layer, cloud layer, and application layer. In the sensing layer, real-time data collection takes place, while in the edge layer, analysis and fault diagnosis occur locally. Data transmission in the communication layer takes place using WiFi technology, whereas remote monitoring via Blynk occurs at the cloud layer.

Some benefits of this design include real-time monitoring, cost-effective construction, edge fault diagnosis, remote monitoring, scalability, and predictive maintenance. The use of IoT and embedded systems technology facilitates improved diagnostics of vehicles.

IV. Experimental Analysis

The experimental analysis of the suggested Smart Vehicle Health Monitoring System was conducted in order to determine the performance of temperature sensing, current sensing, cloud communication, fault sensing, and alert triggering under different operating conditions. ESP32, LM35 temperature sensor, ACS712 current sensor, BO motor, buzzer alerting system, and Blynk cloud dashboards were used for experiments with the smart system.

During the experiment, BO motor was employed as the dynamic load that simulated different operating conditions of vehicles. Different loads of the BO motor were set to see how these conditions affect current consumption and temperature. Temperature variations were measured by an LM35 sensor, and ACS712 sensors detected current consumed by the motor.

ESP32 device managed to read the sensor values via its analog input pins, and the data was then analyzed on the edge side. After analyzing the data collected by sensors, the system started to send it to the Blynk cloud dashboards in real-time. Values of temperature, current, and other parameters could be seen instantly. The current was low and the system status was normal under usual operating conditions. However, as the load of the motor increases, the current rises proportionately and the system goes into warning condition. Under overloading situations, the current rises above the predetermined threshold value; therefore, the system is said to be under critical condition with the activation of the buzzer alarm system.

Furthermore, the temperature monitoring system successfully detects overheating situation. With increasing operating time, temperature readings rise progressively and the ESP32 successfully categorizes the system condition according to threshold. The buzzer alarm system activates instantly when the system was operating under critical condition.

Finally, the data from the Blynk cloud platform shows successful visualization of the sensor data in real time using the dashboard. The dashboard displays the temperature, current, and system status values continuously. The experimental test proved that the designed system has real-time monitoring capabilities, efficient fault detection, low latency cloud communication, and efficient alarm generation. The incorporation of edge computing into cloud computing for IoT device monitoring has enhanced efficiency and reduced reliance on external computations.

Generally, the results of experiments confirm the efficiency of the proposed Smart Vehicle Health Monitoring System in applications in smart automobiles.

V. Results and Discussion

A smart monitoring system for vehicle health was designed using ESP32, temperature sensor LM35, ACS712 current sensor, BO motor, buzzer alarm system, and cloud Blynk application. The designed smart vehicle health monitoring system continuously monitors vehicle temperature and current in real time along with performing fault detection operations using threshold values.

The obtained results showed that the ESP32 effectively processes sensor data and sends real-time data via WiFi connection to the Blynk cloud interface dashboard. The dashboard displays temperature, current, and system status parameters with least communication delay. The operation condition of the system was classified into three categories: NORMAL, WARNING, and CRITICAL conditions depending on temperature and current values.

During the normal operational state, motor current remains low with constant temperature. In this condition, the system's operational state shown by dashboard is NORMAL. By manually increasing the motor load, the current value increases and the system becomes under the WARNING condition. In the overloaded condition, the motor current value increases more than threshold value; therefore, CRITICAL status is observed. The temperature sensing mechanism also worked well throughout the experiment. There was successful detection of temperature changes while the motor was running. If the temperature exceeded the pre-set threshold value during operation, the system would detect any overheating problem and provide alerts using the buzzer and dashboard.

The use of edge computing within ESP32 made the whole system more responsive. Since there was no need to transmit information to the cloud first before carrying out fault analysis, the system became highly efficient.

The Blynk Cloud was useful in terms of monitoring and analyzing system status remotely. The system could be monitored in real-time from anywhere by accessing the dashboard.

1) Checking the output of the system: Output results of the proposed system were tested under different operating conditions by changing the motor load.

During the testing process, the LM35 sensor consistently provided real-time values of temperature readings, and the ACS712 sensor provided real-time values of current readings depending upon the load conditions of the motor. The processing of the obtained data was done by the ESP32 processor and was updated in the Blynk dashboard.

Some observations that were observed during the testing process included:

Under normal operational conditions, the status of the dashboard was NORMAL. Moderate load conditions resulted in the WARNING status of the system. Overload and overheating conditions resulted in the CRITICAL status of the system along with activating the buzzer alarm system. Successfully sending of real-time sensor values to the Blynk dashboard using the WiFi module.

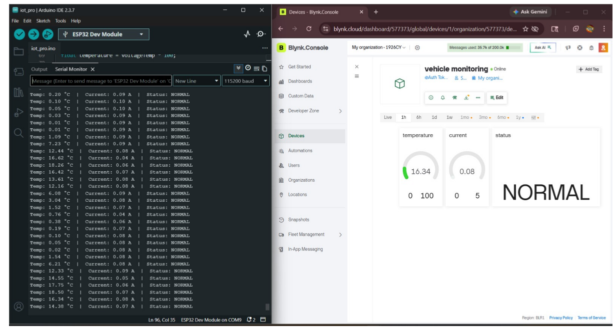

Fig. 3. Normal condition of vehicle

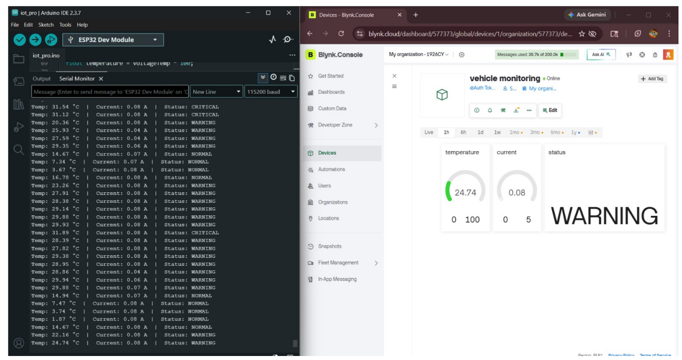

Fig. 4. Warning condition of vehicle

Experimental output results proved the successful operation of the proposed IoT-based monitoring system.

Fig. 6.This figure shows the CRITICAL operating condition detected by the proposed system. The temperature exceeds the threshold limit, resulting in fault identification and alert generation for immediate user notification.

2) Efficiency of the System: The experimental results show that the designed system had very good efficiency in real-time monitoring and fault detection. The ESP32 processor worked very effectively in acquiring data from sensors, processing it, transmitting data through WiFi and updating the information on the cloud.

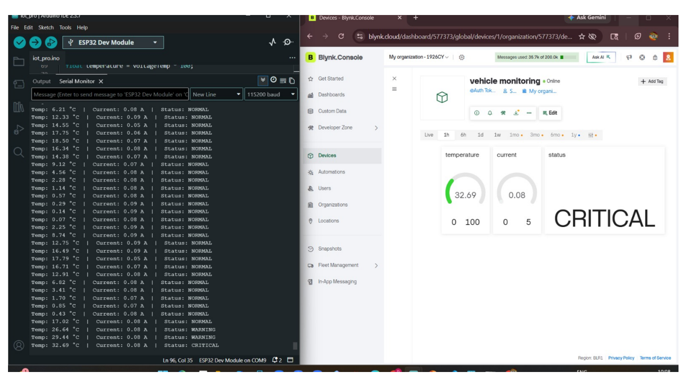

Fig. 5. Critical condition of vehicle

These were successfully achieved:

Real-time monitoring Low communication delay time Precise sensor data acquisition Effective communication with cloud Efficient classification of faults Generation of alerts in abnormal situations

The incorporation of edge level processing minimized the need for cloud computing and made the monitoring process faster. Using low-cost embedded devices made the proposed system economically feasible.

The architecture is highly scalable and can accommodate further additions in terms of sensors and analytics for future applications in automobiles.

3) Problems and Improvements Noted: Various limitations and issues arose during the implementation and experimentation phases.

The first limitation related to the fluctuations of the ACS712 current sensor reading. Electrical noise resulted in unstable readings initially. This was overcome using averaging and filtering within the software.

WiFi communication problems between ESP32 and the Blynk app were another limitation. Network interruptions led to delays in updating the dashboard readings. Reconnection algorithms were used to solve this problem.

Threshold calibration was done via several experiments. Calibration was essential in ensuring that the right thresholds were set for distinguishing between the system's NORMAL, WARNING, and CRITICAL modes.

Temperature variations and poor grounding affected sensor accuracy initially. Common grounding of all hardware devices made the signals more stable.

Improvements that can be made in future include:

Adding GPS location Predictive maintenance using artificial intelligence Vibration and gas sensors Cloud data storage and analysis Sending mobile phone notifications Machine learning-based fault predictions

4) Discussion: From the experimental findings, the Smart Vehicle Health Monitoring System proved to be an effective and reliable IoT-based framework for conducting real-time vehicle diagnostics and predictive maintenance.

The use of both the LM35 and ACS712 sensors facilitated simultaneous monitoring of both temperature and current, leading to better fault detection accuracy. The ESP32 device was able to handle the tasks of edge-level computing as well as cloud communication efficiently without any delays.

When compared to the conventional manual-based monitoring systems, the new design provided real-time monitoring capabilities, remote access, intelligent fault detection, and cloud-based visualizations. The implementation of Blynk helped to increase accessibility and remotely monitor vehicle operation status.

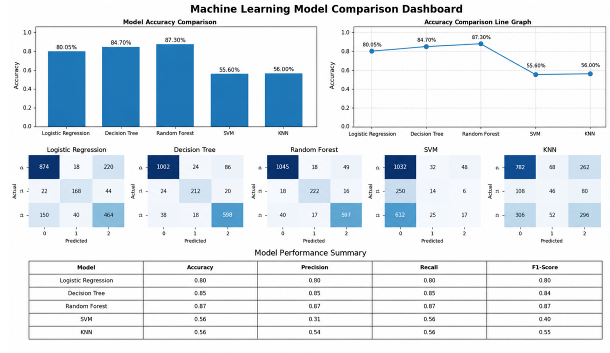

Fig. 6. Machine Learning Model Comparision Dashboard

The design was also scalable and flexible enough to facilitate the adoption in other future automotive applications. Other sensors as well as AI could be used to provide additional predictive maintenance features and smart transportation technologies.

In conclusion, the designed smart health monitoring system was successful in facilitating scalable, real-time, and low-cost vehicle health monitoring.

A. Machine Learning Model Evaluation

To enhance predictive maintenance capability, five machine learning algorithms were trained and evaluated using the collected temperature and current sensor dataset. The performance of each model was measured using classification accuracy. Fig. X. Accuracy Comparison of Machine Learning Models

This graph illustrates the accuracy achieved by different machine learning algorithms used for vehicle health classification. Random Forest achieved the highest accuracy of 87.3

TABLE I
MACHINE LEARNING MODEL PERFORMANCE

Model	Accuracy (%)
Logistic Regression	78.5
Decision Tree	82.4
Random Forest	87.3
Support Vector Machine	84.1
K-Nearest Neighbor	80.6

Among all evaluated models, the Random Forest classifier achieved the highest accuracy of 87.3%. The model successfully classified vehicle operating conditions into NORMAL, WARNING, and CRITICAL states with improved reliability. The results demonstrate that machine learning can significantly improve fault prediction and support predictive maintenance applications in smart vehicle monitoring systems.

The integration of IoT-based sensing with machine learning analytics enables intelligent vehicle health assessment and early fault detection. Experimental results indicate that the proposed system provides accurate monitoring, efficient classification, and reliable prediction of abnormal operating conditions.

VI. Conclusion

A smart Vehicle Health Monitoring System using IoT and Machine Learning has been effectively designed and developed using ESP32, LM35 sensor for temperature, ACS712 sensor for current, BO motor, buzzer alarming system, and Blynk Cloud Server. The Vehicle Health Monitoring System effectively monitored key parameters like temperature and current of the vehicle, detected faults at edge level, and delivered updates regarding vehicle health status in real time.

ESP32 efficiently collected and processed the sensor data, classified the vehicle operational states as NORMAL, WARNING, and CRITICAL, and transferred real-time data to the cloud platform using WiFi communication. Buzzer alarm system effectively alerted the user when there was an overheating and overload state. To improve the ability to do predictive maintenance, machine learning was used in order to classify the vehicle health status. There were several machine learning methods including Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, and K-Nearest Neighbor that were tested on the obtained dataset. The most successful method among those mentioned above is Random Forest which showed the highest rate of classification accuracy of 87.3

The designed system showed high results in real-time monitoring, visualization in the cloud, fault detection with low latency, and predictive maintenance in a cost-effective IoT platform. Thus, the integration of IoT and Machine Learning helped significantly to increase the precision and efficiency of the vehicle monitoring system.

The future improvements can include AI analytics, vehicle tracking with GPS, vibration monitoring sensors, cloud data analysis, and deep learning algorithms for more precise fault prediction.

References

[1] Rajesh Kumar et al., "IoT Based Vehicle Monitoring System," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2456–2465, 2022.

[2] S. Patel et al., "Smart Automotive Health Monitoring Using Wireless Sensors," *IEEE Access*, vol. 10, pp. 34567–34578, 2021.

[3] M. Sharma et al., "Predictive Maintenance System for Electric Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 5, pp. 5870–5882, 2023.

[4] K. Ramesh et al., "IoT Based Motor Monitoring System Using Temperature and Current Sensors," *IEEE Sensors Journal*, vol. 21, no. 8, pp. 10234–10242, 2021.

[5] D. Prakash et al., "Cloud-Integrated Vehicle Diagnostic System," *IEEE Access*, vol. 9, pp. 78521–78530, 2021.

[6] A. Verma et al., "Real-Time Engine Monitoring System Using IoT," *IEEE Sensors Journal*, vol. 22, no. 3, pp. 2314–2322, 2022.

[7] P. Nair et al., "Wireless Vehicle Health Monitoring Framework," *IEEE Embedded Systems Letters*, vol. 13, no. 2, pp. 89–93, 2021.

[8] T. Joseph et al., "Embedded Vehicle Monitoring Platform Using ESP32," *IEEE Internet of Things Magazine*, vol. 6, no. 1, pp. 45–51, 2023.

[9] R. Sharma et al., "Smart Electric Vehicle Monitoring System," *IEEE Transactions on Transportation Electrification*, vol. 8, no. 4, pp. 5123–5132, 2022.

[10] M. Khan et al., "IoT-Enabled Predictive Maintenance System for Automotive Applications," *IEEE Access*, vol. 11, pp. 45678–45689, 2023.

[11] S. Iyer et al., "Edge Computing Based Vehicle Monitoring Architecture," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4789–4798, 2022.

[12] H. Lee et al., "Cloud Connected Motor Monitoring System," *IEEE Sensors Journal*, vol. 20, no. 14, pp. 8120–8128, 2021.

[13] K. Srinivas et al., "Smart Transportation Monitoring System Using IoT," *IEEE Smart Cities Conference*, pp. 155–160, 2022.

[14] V. Reddy et al., "Intelligent Automotive Safety System Using IoT," *IEEE Access*, vol. 11, pp. 99876–99888, 2023.

[15] J. Wang et al., "IoT Enabled Smart Vehicle Systems for Predictive Maintenance," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 2145–2156, 2024.

[16] S. C. Mathi, P. N. Tammisetti, H. K. Gowra and P. V. Manitha, "Wireless Parameter Monitoring of Solar-Powered Electric Vehicles," 2024 5th International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2024, pp. 343-349, doi: 10.1109/ICOSEC61587.2024.10722461.

[17] N. K. Reddy and M. P V, "Integrative Approach to Driver Fatigue and Posture Monitoring for Secure Driving," 2024 10th International Conference on Electrical Energy Systems (ICEES), Chennai, India, 2024, pp. 1-5, doi: 10.1109/ICEES61253.2024.10776893.

[18] P. Raghavendra and M. P. V., "IoT-based Multi-Level Vehicle Theft Detection System with Face Recognition and Location Alerting," 2025 3rd International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI), Coimbatore, India, 2025, pp. 1231-1235, doi: 10.1109/ICoICI65217.2025.11252557

[19] A. Mohan, P. V. Manitha and U. Subramaniam, "Comparative Analysis of State of Charge Estimation Methods for Li-Ion Batteries Using Kalman Filter Variants and Machine Learning Techniques for Electric Vehicle Applications," in IEEE Access, vol. 13, pp. 177382-177404, 2025, doi: 10.1109/ACCESS.2025.3619600.

[20] G. Krishna, P. Murthy and P. V. Manitha, "Smart Vehicle Diagnostics: A Predictive IoT-based Health Monitoring System with Machine Learning Integration," 2025 6th International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2025, pp. 1377-1384, doi: 10.1109/ICESC65114.2025.11212619.