Intelligent Vehicle Monitoring System

Ashwin Pillai[1], Tanmay Salavkar[2], Srushti Wategaonkar[3], Harsh Rao[4] and Deepa Ekhande[5]

[1,2,3,4,5]Department of Computer Engineering, Pillai College Of Engineering (PCE), University of Mumbai ,New Panvel, India.
ashwina22comp@student.mes.ac.in

Abstract. With the advent of Internet of Things (IoT) technologies and On-Board Diagnostics (OBD) systems in modern vehicles has enabled continuous real-time data collection from vehicular sensors paving new way for predictive maintenance. In this paper, we have summarized architectures of various recent research papers by assessing the way they get vehicular parameters-either using OBD-II data or external sensors, preprocessing techniques applied on those data and how they gained useful insights for the data. Apart from traditional architecture, this paper also proposes a newer architecture, to enhance data collection, onboard vehicle maintenance and secure transmission of data to the cloud.

Keywords: Internet of Things (IOT), Onboard Diagnostics (OBD), Machine Learning (ML), Fleet Management, Auto encoders, Predictive Maintenance, Vehicle Monitoring, CAN bus.

1 Introduction

Modern vehicles enable technicians to track the vehicle health using diagnostic codes emitted by advanced and sophisticated ECUs (Electronic Control Units) on the CAN bus, using proprietary tools, proprietary to the manufacturer. Thus OBD-II standard was introduced to cover all common PIDs representing common problems under a common standard. It includes parameters like Engine load, Fuel pressure, RPM, Speed, etc. These parameters are accessible using the OBD port in the vehicle.
In recent years, people tend to access these predefined OBD-II parameters to gather data using tools like ELM327 or MCP2515 along with microcontrollers like ESP32, etc for analyzing vehicle parameters. Then traditional supervised methods, like Support Vector Machines (SVM) and Random Forests are applied, showing strong predictive performance [7], but they usually require labeled datasets for training the machine learning models. But unsupervised learning methods designed for streaming data, such as TEDA-RLS are becoming popular because they can spot outliers without needing labeled data, making them well-suited for real-world automotive applications that benefit from incremental learning [13]. Usage of IOT devices like GSM and GPS enabled to locate each vehicle in the fleet and transmission of vehicle parameters over the internet either to keep analyzed data or offset analysis to cloud servers for detecting future requirements of maintenance [14]. These advances have helped develop smart

P. Goyal et al. (eds.), *Proceedings of the Conference on Technologies for Future Cities (CTFC 2025)*, Atlantis Highlights in Sustainable Development 7,
https://doi.org/10.2991/978-94-6239-650-0_8

vehicle monitoring systems focused on reducing maintenance costs, improving safety, and extending vehicle lifecycles [3].
This survey takes a closer look at the existing architecture to process vehicle parameters, summarizing each and every architecture along with their potential limitations.

2 Literature Review

In the paper by Christy Mary Jacob et al. [1] automates traffic violation monitoring, transmission and reporting to RTO linked to respective vehicle by tracking offenses like over speeding, rash driving, drunk driving, and seat belt violations from the moment the vehicle starts using two modes, automatic and manual [1]. To monitor and control speed of vehicle, proximity sensor is used by automatic mode [1] whereas, manual mode uses RFID card to authorize driver to start vehicle, by checking driver's associated licensing with server [1].
Whereas, S Sethuraman et al. [2] explains about a vehicle tracking system that observes the performance of the vehicle and driver behavior. The study uses a USB camera, alcohol sensor, inclinometer and GPS sensor connected to a raspberry pi board [2]. When the vehicle is started, crash is detected using piezo electric pulse or any changes to inclination beyond threshold value, a picture of driver along with location is recorded and sent through mail and SMS [2].
Similarly in the paper by M S Punith et al. [3] have described a smart fleet management system which makes use of different sensors measuring battery temperature, seat occupancy (for driver detection), battery voltage, tire pressure (keeping above 25%), brake fluid, coolant level, engine temperature, camera and fuel level. These sensors, data are collected and transmitted to cloud servers where data is processed, analyzed and further displayed on a cloud dashboard which can be accessed using a smartphone with proper authentication [3]. Twilio API is used to send alert notifications to the user [3].
But Mahaalakshmi B et al. [4] being focused on exploring machine learning models on vehicle data, proposed the Smart Vehicle health monitoring System that uses Ordinal Logistic Regression model which provides or predicts the health of the car as “Good”, “Moderate” or “Bad” condition. It also predicts some parameters which are achieved using the ARIMA model [4]. Health of vehicles and predictions of parameters are displayed on a web application along with monthly reports [4].
In a similar way, Pavan Chandra Vishal Chaganti et al. [10] integrated Arduino uno board with sensors like accelerometer, ultrasonic sensor, GPS module, heat sensor, and for detecting changes in acceleration, obstacle detection, location tracking,
Similarly, Neil Johnson et al. [12] presented a method for predictive maintenance by collecting data from various vehicle sensors, cleaning, normalizing, and transforming them into structured data consisting of eleven attributes. The authors developed a five-layer DNN model using RELU activation functions in its hidden layers and SoftMax in the final layer to categorize vehicle health into 4 classes: Excellent, Good, Needs Service, and Very Bad [12]. The model is trained and then tested with an accuracy of 97.68%, outperforming Random Forest model with accuracy of 91.24% [12]. Protection of temperature sensitive items, respectively [10]. Three attributes, namely,

temperature, distance, and speed were used to categorize them into two categories, namely, safe or dangerous. Here, the Random Forest algorithm performed better than SVM [10].

But a different approach was used by D. Siswanto et al. [9]. It includes, performing literature review on maintenance procedures, performing a needs analysis to determine required system functions, creating a data flow diagram to illustrate data processes, and designing a web database using MySQL® to control data effectively. Architecture involves Arduino with GPRS capabilities connected to OBD device via Bluetooth and transmission of data to IOT gateway. A web dashboard contains a deployed web server, APIs for data transmission between the web interface and vehicle sensors (using IOT gateway), stored in MySQL® database and showing maintenance status of vehicles. Rule based approach is used, for instance, checking the number of kilometers run by vehicle or setting periodic intervals to predict vehicle maintenance required for certain parts of vehicle [9].

Similarly, More et al. [11] presented AutoShield which collects OBD data from ELM327 to mobile phones. After user registration is completed and connect their vehicles to the platform which fetches engine speed fault and position data. This is cleaned, processed and analyzed using ml models to provide detailed insights on driving ethics, etiquettes and vehicle efficiency. Features like geofencing, traffic-aware fuel station routing and personalized car interior recommendations add value. The data is securely stored in the cloud when the mobile has a reliable and secure internet connection ensuring scalability and user interaction through dashboards and alerts [11].

In the paper, Babiyola et al. [5] focuses their architecture on visualization of data from various sensors (connected to Raspberry Pi SBC) like MEMS sensor [5], camera sensor, fuel level sensor, accelerometer, Tire pressure sensor and GPS module (for geofencing). Six step approach for implementation: at first, the fleet is chosen. According to the type of fleet, necessary hardware and software components required are decided.[5] Then all components are integrated along with functional testing. A web interface is used for real-time monitoring with prompt alerts using SMS [5]. But analysis is limited to predefined rules or intervention of human [5].

But, the authors Jegan Pranav et al. [6], described a fleet management system that uses IoT technology to monitor fuel levels using ultrasonic sensors, detect theft and track vehicles in real-time using RFID and GPS modules integrated with ESP8266 respectively for each fleet. For each vehicle, all these data will be sent to fleet owners' servers through each vehicle's gateway. Then, the data is processed and stored in the Thingspeak cloud along with MATLAB® for analysis [6].

Similarly, Sheshang et al. [8], implemented ESP8266 based IoT sensors using weight sensors and GPS helping them in monitoring the payload weight and precise location tracking, which is transmitted to a central server via WIFI [8]. The system saves historical data for report generation and performance analysis for every vehicle and ensures scalability, supporting addition of more vehicles. It also includes remote engine shutdown to counter theft and geofencing for setting operational boundaries making the fleet more secure [8].

Table 1. Literature Survey Table

Ref. No.	Limitations
[1]	1. External Sensors can cause calibration issues in varied environment conditions. 2. Checking the vehicle speed is done by a proximity sensor, even though this function can be more accurately and efficiently done by the OBD2 scanner itself, eliminating the need for this sensor.
[2]	1. Monitoring is limited to only burglary detection. 2.Piezoelectric pressure sensor is inaccurate for detecting crashes. Need to use a combination of sensors like gyroscope, accelerometer and microphone for detection of crash.
[9]	1.The vehicle data is transmitted using unencrypted network packets transmission method like HTTP. 2. Decision Support System categorized as rule based is used which could miss out representative predictions in real world.
[3]	1. Demonstration of implementation is done using potentiometers instead of onboard vehicle sensors which could show values representative to the real world. 2.Implementation limited to data visualization.
[4]	1.Categorises the prediction with only three classes- Good, moderate and bad. 2.The prediction is not specific. 3. Only six basic parameters are used for prediction.
[11]	1. Use of mobile phones to gather and process data and to keep it connected to OBD, throughout the vehicle operation. Chances of data corruption while Bluetooth transmission is possible. 3. ELM327 transmits unencrypted packets, leading to security issues as it could be sniffed by anyone in the vicinity [15].
[5]	1.Use of the camera to detect driver behavior is computationally expensive for a single board computer especially while accident prevention, which is also processing other sensors data simultaneously. 2. The paper does not implement the use of AI along with its architecture. 3.The mechanism for shutting the engine, in case of owners knowing potential theft of a vehicle is not clear.
[6]	1.There is no mechanism to handle data transmission in case of network unavailability. 2.The ultrasonic sensor is used to detect fuel level instead of using OBD (Onboard Diagnostics). 3. It follows a rule-based approach for data analytics.

[8]	1.It does not provide enough details about how data transmission is kept secure. 2.The system relies entirely on ESP8266 Wi-Fi for sending data instead of using reliable cellular networks. 3. Since it uses simple rule-based logic, the system can't effectively tell real events apart from false alarms in certain cases, which could lead to too many unnecessary notifications.
[10]	1. Use of external sensors instead of vehicle sensor data using obd. 2.Predicts only two classes as safe or dangerous, limiting users for better insights from vehicles.
[12]	1. The model showed difficulty in distinguishing good and very bad health states [12]. 2. Prediction limited to only four classes namely; excellent, good, needs service, and very bad. 3. Limited high-quality labeled data lowers effectiveness to training and reduces predictive accuracy.

3 Proposed System

In the existing OBD systems, in literature, Espressif microcontroller boards were used which were paired with ELM327. Due to limited clock speed and memory, data acquisition and detection are limited to three parameters according to paper [10]. ELM327 uses Bluetooth and it causes security and privacy issues due to it transmitting unencrypted packets [15]. Instead, we connect the OBD port of the car through a wired connection, interfaced using MCP2515 IC and MCP2551 transceiver. Secure transmission of fleet owners' vehicles could be done using the GSM module with a web server. Using onboard WIFI chips present in SBC, data about vehicular parameters could be visualized through encrypted communication for mobile visualization.

3.1 Workflow

The proposed system architecture, as shown in Fig. 1, is designed for intelligent vehicle monitoring and predictive maintenance using machine learning and IoT integration. The process begins with vehicle sensors that emit real-time data through the CAN (Controller Area Network) bus.

This data is captured by a single board computer, which acts as the central processing hub. The captured data undergoes initial preprocessing and filtering to remove noise and ensure the quality of inputs before being fed into a trained machine learning (ML) model. This ML model, previously trained using a dataset formed from earlier acquisition and preprocessing phases, is deployed on the same embedded system. The model generates predicted outputs which could be displayed on a driver-facing LCD module for real-time insights or using a beep or flash, in case a mobile phone is not present for visualization.

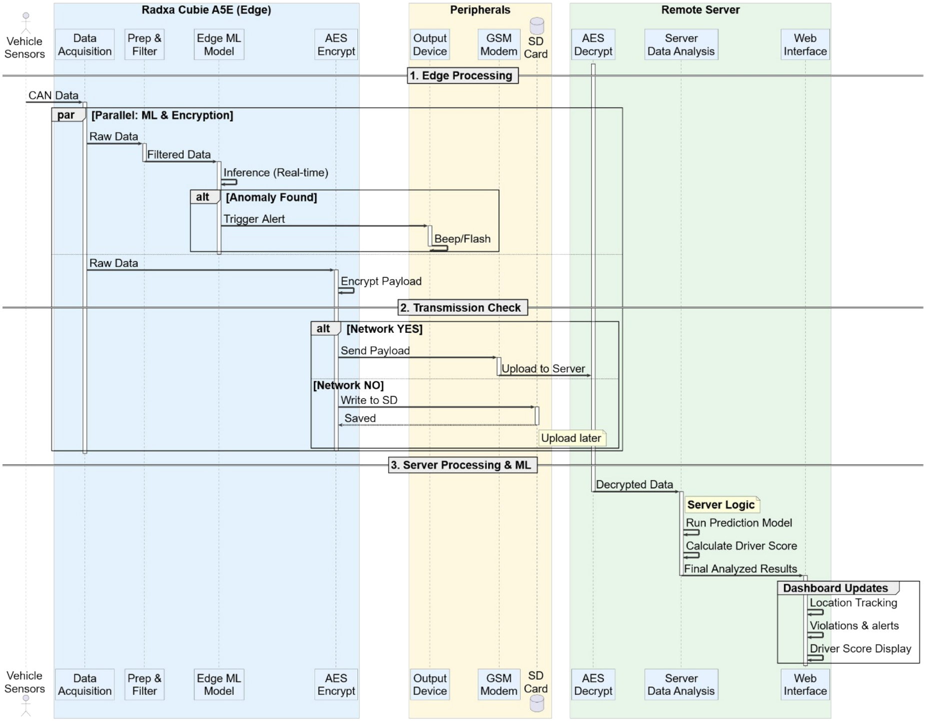

Fig. 1. Proposed System Architecture

Simultaneously, the raw or compressed sensor data is written to an SD card. Once network is available, the data is transmitted to remote server, else stored locally. The server-side system conducts further data analysis when requested by admin, and the results are visualized through a web interface which supports these major functions like location, violations tracking, driver score and vehicle anomaly detection. The driver score is calculated using this formula:

final_driver_score = 100 - (scaled composite score) = max (0, 100 - (composite_score * scaling_factor)) (1)

composite_score = (freq_high_rpm + freq_speeding + freq_high_accel_pos + freq_high_engine_load) (2)

Each represents percentage of respective parameters with instance of values higher than threshold:

Table 2. Parameter Threshold Table

Parameter	Threshold
RPM	3500
ACCELERATOR_POS	80%
SPEED	100

ENGINE_LOAD	80%

scaling_factor = 100 / (number_of_metrics * acceptable_bad_percent) (3)

Here, scaling factor is set sensitivity for making penalties less or more severe. The number of matrices represents number of parameters chosen for composite score calculation and the acceptable_bad_percent could be tweaked by the admin itself. e.g., 100 / (4 * 5) = 5, since scaling factor is 5 and if 5% "bad" in all 4 categories, so final score will be zero.

The DBSCAN algorithm groups data points that are closely packed together, marking as outlier points that lie alone in low-density regions. It is efficient, especially with spatial indexing, making it highly suitable for on-device applications [24]. Thus, this algorithm is ideal for memory and compute constraint IOT devices. Every vehicle has a different ECU (Electronic Control Unit). The OBD PIDs are specific to each control unit respectively. Thus, for each group of PIDS corresponding to respective control units, clusters for each will be formed. When the new data point is away from the cluster, the model would predict that there is anomaly in the car using that cluster. It could be indicated in LCD (Liquid Crystal Display) which will be used to show the region that car is likely affected or it could be due to driver's driving behavior.

Whereas, Autoencoders are neural networks that learn to compress and reconstruct data, are mainly used for dimensionality reduction and anomaly detection [25]. They require a computationally intensive training process [25]. Thus, uploaded data would be used to train and reconstruction of data is done to check error in the predicted data with actual data to find the anomaly in car.

3.2 Components

MCP2515 with automotive grade TJA1050:. As depicted in Fig 2, the MCP2515 module is a standalone controller which handles low-level tasks of receiving and transmitting messages on the CAN bus via OBD-II port. It consists of MCP2515 chip itself along with TJA1050 CAN transreceiver. TJA1050 acts like a physical interface between digital signals from MCP2515 chip and differential voltages by CAN bus. TJA1050 requires 5V, so, mcp2515 module is supplied with 5V from pin 2 of Radxa board [26]. Level shifting is done between MISO (Master In [Radxa board] Slave Out [MCP2515]) pin of MCP2515 module and pin 38 of Radxa board. The module communicates with Radxa board, feeding it the continuous stream of vehicle data. For this project, the mcp2515 module is connected to spi1 of a single board computer and it appears as '/dev/spidev1.0'. Using the 'spidev' python package, we communicate with MCP2515 to get vehicle parameters.

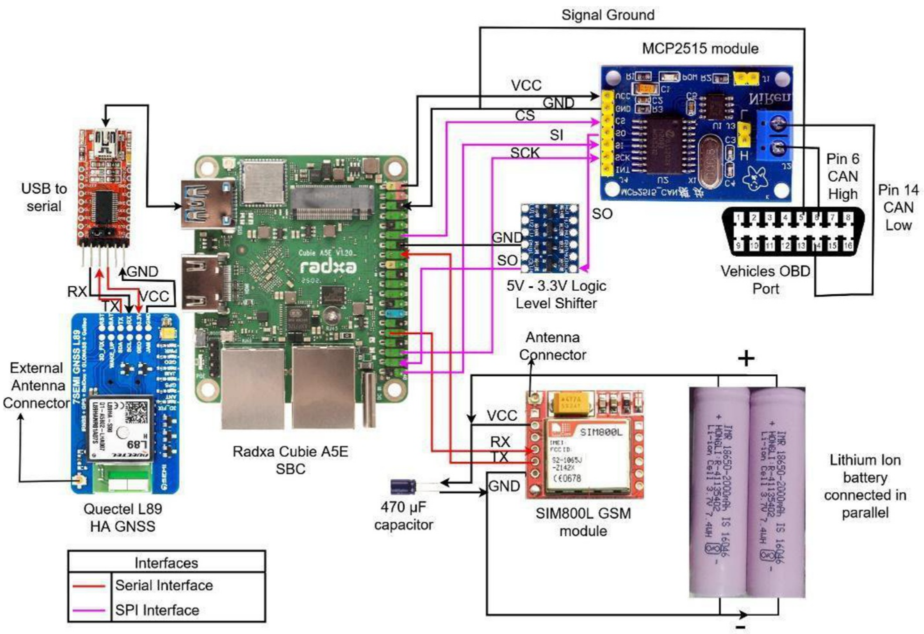

Fig. 2. Hardware Interconnection [26]

Radxa Cubie A5E:. It is a powerful and adaptable single-board computer (SBC) responsible for all onboard tasks built around an octa-core ARM Cortex-A55 Allwinner A527 processor. It provides a flexible 1GB-4GB LPDDR4 RAM package and robust storage via micoSD/EMMC/NVME (2230 form factor). It contains onboard Wifi 6 and bluetooth 5.4. Here, it acquires raw CAN bus data from MCP2515 and preprocesses the data. It also executes machine learning models in real-time for predictive maintenance. Moreover, it manages GSM module's cellular communication making sure that data uploads to remote servers.

Quectel® L89HA:. GNSS module such as Quectel® L89HA is integrated in Fig. 2, to enable location tracking of the vehicle. Signals from multiple satellites is received to determine vehicle's precise location coordinates. It supports multiple satellite constellations. The latest NMEA sentence is chosen from '$GNGGA', '$GPGGA', '$GLGGA', '$GAGGA', '$GBGGA', '$GIGGA' and is parsed using pynmea2 library. This location data is passed to the Radxa board, which matches it with the vehicle's operational data. The combined data is then transmitted to the server, allowing for real-time location tracking and analyzing historical routes. This module is connected to FTDI FT 232 RL which is connected to the USB port.

SIM800L®:. SIM800L® is a GPRS/GSM module which requires a decoupled layout according to Fig. 2 due to its power requirement of 3.7V – 4.2V with peak current of 2A during data transmission. A capacitor is added between VCC and GND to compensate for the sudden energy demand during transmission of data. It is used to establish a connection to a 2G network using a standard SIM card to connect to remote server. Since this module's application layer support only HTTP, so AES encryption is used for encrypting data payload in SBC itself, and then this payload is sent to SIM800L® using AT commands and then to server. Here the SIM800L® is connected to serial port of single board computer via GPIO headers.

FTDI FT 232 RL:. It is a USB to serial convertor module capable of providing logic levels of either 3.3 volts or 5 volts. It converts USB protocols into communication language understandable to modules compatible i.e. serial communication and vice versa. In Windows®, this module appears as COM* port and in Linux®, it appears as /dev/ttyUSB*.

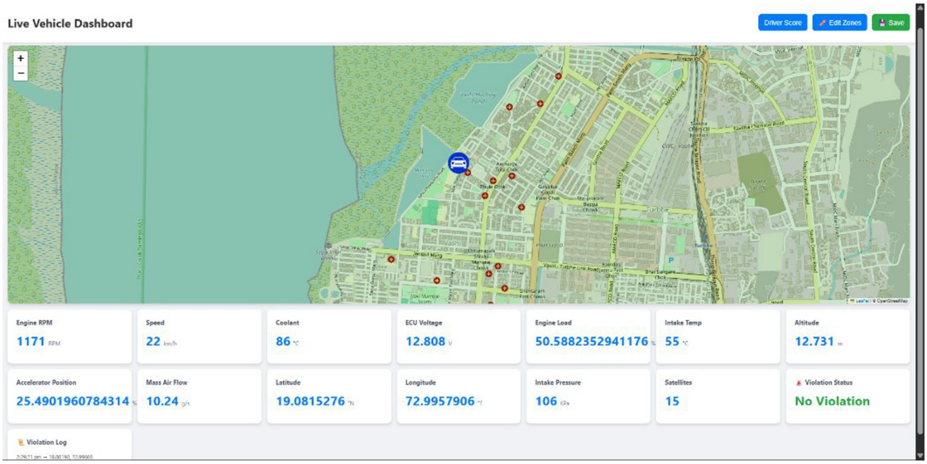

Fig. 3. Realtime Vehicle Dashboard

4 Results

The OBD parameters uploaded using Sim800L® are sent to a server securely and the realtime data is viewed by admins having dashboard access as shown in Fig. 3. Geofencing features like allowed zone and restricted zone could be added by admins and in case of any violation, the data is logged for that driver. Quectel® L89HA worked better than U-blox® Neo-6M since it gave accurate coordinates even in presence of buildings in the surroundings and could simultaneously take GPS data from more than 30 satellites (if available). The dataset includes the following parameters: timestamp, RPM, SPEED, COOLANT_TEMPERATURE, ENGINE_LOAD, INTAKE_TEMP, MAF (Mass Air Flow), INTAKE PRESSURE, CONTROL_MODULE_VOLTAGE, and ACCELERATOR_POS. The following correlation matrix, Fig. 4, represents the parameters as same as the above order.

The autoencoder model used has- 1 Input layer + 5 Dense layers (activation = relu each) + 1 Output layer (activation = linear) with Hyperparameters: EPOCHS = 80, BATCH_SIZE = 256, LATENT_DIM = 6, PATIENCE = 8. The anomaly detection model (using Autoencoder) was trained on a dataset containing features like diff_features (derived from raw data). Fig. 4 shows the correlation of each relevant raw features collected from car.

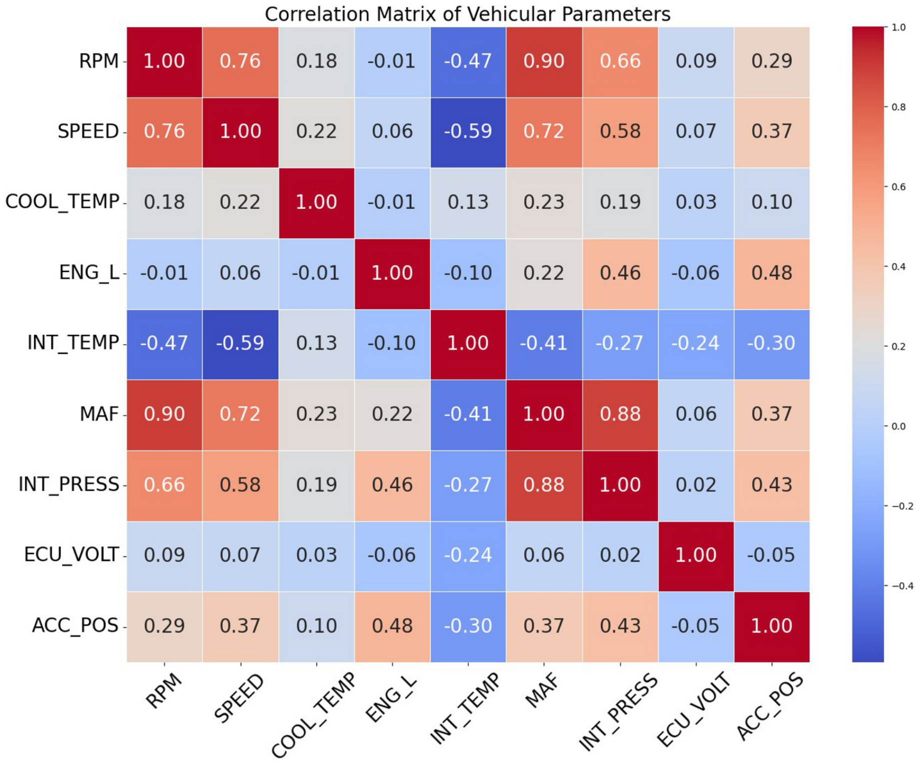

Fig. 4. Correlation between parameters

Diff_features measure the change in value of corresponding vehicle parameters, and reconstruction_mse, indicating how unusual a row is compared to the learned normal behavior. The model is configured as an autoencoder with a latent dimension of 6. Out of 20,000 data points (approx. 15 hours driving), only 279 were flagged as anomalies, but these instances were inconsistent and can be ignored, as shown in Fig. 5. Additionally, values exceeding the threshold might be influenced by normal driving behavior, so they too can be disregarded. Therefore, we could conclude that there are no significant anomalies in the dataset. For detecting true mechanical issues in vehicles, larger MSE values (e.g., 2, 3, 8, 10) sustained over longer periods, like a day or an extended trip, would be required to indicate an actual anomaly.

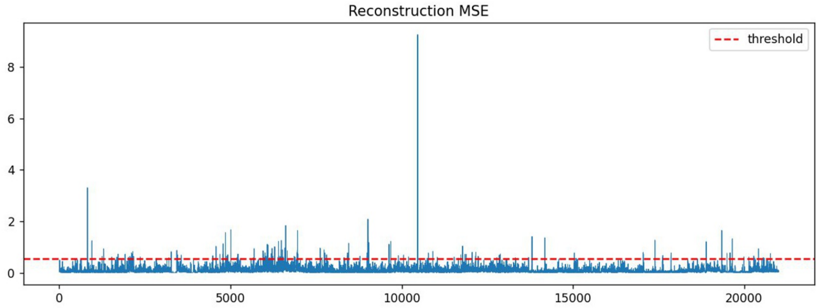

Fig. 5. Graph that was viewed at admin dashboard (Anomaly vs Timestamp)

5 Conclusion

The above system presents the use of high-performance low-cost single board octa-core computer integrated with its onboard WIFI, external MCP2515 module, SIM800L® GSM module and Quectel® L89HA GNSS module, could pose as a possible replacement for unsecure ELM327[15] and for low performance and tightly constrained onboard memory device like ESP32. The current system tells anomaly in the car but granularity towards the anomaly is limited. E.g.- In Air & Fuel System, PIDs like MAF, Intake Pressure, Engine Load, RPM indicating about air–fuel mixture and combustion health could detect issues like Clogged air filter, fuel imbalance or faulty sensors itself. Thus, fleet owners could manage the activity and rate drivers of corresponding vehicles using a web dashboard.

References

1. C. M. Jacob, N. George, A. Lal, R. J. George, M. Antony and J. Joseph, "An IoT based Smart Monitoring System for Vehicles," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184), Tirunelveli, India, pp. 396-401, doi: 10.1109/ICOEI48184.2020.9142936, (2020)
2. S. Sethuraman and S. Santhanalakshmi, "Implementing Vehicle Black Box System by IoT based approach," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184), Tirunelveli, India, pp. 390-395, doi: 10.1109/ICOEI48184.2020.9142906, (2020)
3. M. S. Punith, M. Nithya and K. Deepa, "IoT Enabled Smart Fleet Management," 2022 IEEE 4th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA), Goa, India, pp. 256-260, doi: 10.1109/ICCCMLA56841.2022.9989097, (2022)
4. M. B, P. R, E. J. S. J and C. S, "Smart Vehicle Health Monitoring System Based On ARIMA and Ordinal Logistic Regression for Analysis of Vehicle Parameters," 2023 International Conference on System, Computation, Automation and Networking (ICSCAN), PUDUCHERRY, India, pp. 1-6, doi: 10.1109/ICSCAN58655.2023.10395495, (2023)
5. A. Babiyola, V. Saillaja, S. P. Shally and S. Omkumar, "Development of an Internet of Things-based Integrated System for Fleet Management in RealTime," 2023 2nd

International Conference on Edge Computing and Applications (ICECAA), Namakkal, India, pp. 1265-1269, doi: 10.1109/ICECAA58104.2023.10212331, (2023)
6. J. Pranav.P, A. Jenifer.J, S. Janish.A, N. K and J. Ernest.C, "Smart Fleet: An Advanced Web-Based Solution for Fleet Management and Security," 2023 International Conference on Circuit Power and Computing Technologies (ICCPCT), Kollam, India, pp. 530-535, doi: 10.1109/ICCPCT58313.2023.10245993, (2023)
7. Kumar R, Jain A. Driving behavior analysis and classification by vehicle OBD data using machine learning. J Supercomput., doi: 10.1007/s11227-023-05364-3. Epub ahead of print. PMID: 37359337; PMCID: PMC10198028, (2023)
8. S. Degadwala, R. Upadhyay, S. Upadhyay, S. S. Dave, D. Mahida and D. Vyas, "Enhancing Fleet Management with ESP8266-based IoT Sensors for Weight and Location Tracking," 2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bengaluru, India, pp. 13-17, doi: 10.1109/ICIMIA60377.2023.10425949, (2023)
9. D. Siswanto, A. Farid, G. Priyandoko and N. A. Wahyudi, "An IoT-Based Decision Support System for Real-Time Vehicle Maintenance Scheduling in the Fourth Industrial Revolution," 2024 Beyond Technology Summit on Informatics International Conference (BTS-I2C), Jember, East Java, Indonesia, pp. 687-692, doi: 10.1109/BTS-I2C63534.2024.10942271, (2024)
10. P. C. Vishal Chaganti, M. P. V. and M. Nithya, "Fleet Management Using IoT and Machine Learning Techniques," 2024 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, pp. 498-503, doi: 10.1109/ICICV62344.2024.00084, (2024)
11. A. More, S. Khane, D. Jadhav, H. Sahoo and Y. K. Mali, "Auto-shield: Iot based OBD Application for Car Health Monitoring," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, pp. 1-10, doi: 10.1109/ICCCNT61001.2024.10726186, (2024)
12. N. Johnson, S. E. V. Ewards, S. Silas and G. J. W.Kathrine, "Predictive Vehicle Maintenance using Deep Neural Networks," 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS), Coimbatore, India, pp. 322-326, doi: 10.1109/ICC-ROBINS60238.2024.10533924, (2024)
13. P. Andrade, M. Silva, M. Medeiros, D. G. Costa and I. Silva, "TEDA-RLS: A TinyML Incremental Learning Approach for Outlier Detection and Correction," in IEEE Sensors Journal, vol. 24, no. 22, pp. 38165-38173, doi: 10.1109/JSEN.2024.3458917, (2024)
14. S. Jeevan, K. K. S, J. Sharma, Mohana, M. Moharir and A. K. A R, "GPS Based Efficient Real Time Vehicle Tracking and Monitoring System Using Two Factor Authentication and Internet of Things (IoT)," 2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI), Coimbatore, India, pp. 473-478, doi: 10.1109/ICoICI62503.2024.10696765, (2024)
15. MITRE, “CVE-2019-12797,” MITRE Corporation, 2019. [Online]. Available: https://cve.mitre.org/cgi-bin/cvename.cgi?name=2019-12797, last accessed 2025/12/12
16. Prajakta P. Deshpande, “An IoT based Fleet Management System for School Buses”, International Journal of Engineering Research & Technology (IJERT), Vol. 10 Issue 10, (2021)
17. Muntimadugu, et al., “IoT-based Web-Integrated OBD-II Telematics for Real-Time Vehicle Health Monitoring System”. 91-96. 10.1109/iTechSECOM59882.2023.10435307, (2023).
18. A. Nagarajan et al. “Fleet Management System Using IoT”, International Journal of Creative Research Thoughts, Volume 8, Issue 4, IJCRT2004065, (2020)
19. R.Mahalakshmi et al. (2019). “Fleet Automation Using IoT Logistics. International Journal of Engineering and Advanced Technology”. 8. 4231-4238. 10.35940/ijeat.F8911.088619.

20. Şirin, FO et al., “Development of Fleet Tracking Systems Using Direct CAN Data”. European Journal of Science and Technology (39), 44-50. https://doi.org/10.31590/ejosat.1144037, (2022)
21. G. Ramesh Babu et al. “Development of IoT Based Logistic Vehicle Maintenance System”, Dogo Rangsang Research Journal, Vol-09 Issue-01 No. 01, (2022)
22. Singaraju, NagaLakshmi., Real-Time Fleet Tracking and Diagnostic System Using CAN Bus and GPS. International Journal on Science and Technology. 16. 10.71097/IJSAT.v16.i3.6895, (2025)
23. K. M and A. K. V, "IoT-based Web-Integrated OBD-II Telematics for Real-Time Vehicle Health Monitoring System," 2023 International Conference on Intelligent Technologies for Sustainable Electric and Communications Systems (iTech SECOM), Coimbatore, India, 2023, pp. 91-96, doi: 10.1109/iTechSECOM59882.2023.10435307, (2023)
24. Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96). AAAI Press, 226–231, (1996)
25. G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," Science, vol. 313, no. 5786, pp. 504-507, doi: 10.1126/science.1127647, (2006)
26. Radxa Documentation, "40 Pin GPIO Interface," Radxa Cubie A5E Hardware Use. [Online]. Available: https://docs.radxa.com/en/cubie/a5e/hardware-use/pin-gpio , last accessed 2025/12/12