IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2026

DOI: 10.17148/IARJSET.2026.13112

Driver Identification and Activity Tracking with Geo-Fencing & Number KM Lock Using Python and Raspberry Pi

Ujwala B S[1], Chandan Ganesh Gouda[2], C G Poorvi[3], Bhuvan Kumar R[4], Chinmayi S[5]

Assistant Professor, Dept. of Electronics and Communication Engineering,

JNN College of Engineering Shimoga, India[1]

UG Scholar, Dept. of ECE, Jawaharlal Nehru New College of Engineering,

Shimoga, Karnataka, India, USN :4JN22EC023[2]

UG Scholar, Dept. of ECE, Jawaharlal Nehru New College of Engineering, Shimoga, Karnataka, India,

USN :4JN22EC022[3]

UG Scholar, Dept. of ECE, Jawaharlal Nehru New College of Engineering, Shimoga, Karnataka, India,

USN :4JN22EC020[4]

UG Scholar, Dept. of ECE, Jawaharlal Nehru New College of Engineering, Shimoga, Karnataka, India,

USN :4JN22EC027[5]

Abstract: This paper presents an enhanced intelligent vehicle monitoring and security system integrating multi-factor driver authentication, real-time GPS tracking, geo-fencing, and distance-based vehicle locking using Raspberry Pi and Python. Unlike traditional systems, the proposed model incorporates biometric fingerprint matching, face-recognition-based driver identification, and single-bit authentication to activate vehicle ignition. The system continuously collects GPS and sensor data to monitor vehicle movement, enforce geo-fence boundaries, and automatically restrict operation when predefined KM limits are exceeded. AI-enabled modules further support driver activity prediction and periodic in-drive re-authentication to prevent unauthorized access, misuse, and safety risks. Experimental implementation confirms reliable performance across real-time tracking, identity verification, geo-fence breach response, and automated lock mechanisms, demonstrating strong potential for applications in fleet monitoring, rental automation, and intelligent transport management.

Keywords: Fleet Management, Theft Prevention, GPS Tracking, Real-time Monitoring, Microcontroller Automation.

I. INTRODUCTION

In today's rapidly evolving transportation and logistics sectors, the need for secure and efficient vehicle management systems has become increasingly critical. With the rise in vehicle usage across personal, commercial, and industrial domains, ensuring the safety and proper utilization of vehicles is a growing concern. One of the major challenges lies in preventing unauthorized access and curbing vehicle theft or misuse. Conventional vehicle security systems often fall short when it Recent advancements in embedded intelligence comes to real-time monitoring and control, making them inadequate for modern demands.[1]

The demand for intelligent vehicle tracking systems that can provide immediate data, alerts, and control mechanisms is steadily increasing. Organizations and individuals alike are seeking solutions that offer seamless driver identification, reliable activity logging, and contextual restrictions such as geo-fencing and kilometer-based usage limits. These requirements underline the necessity for a system that not only detects anomalies but also proactively manages them in real-time.[2] have enabled vehicles to integrate multi-layered authentication and real-time analytics for improved safety. Modern security systems now rely on biometric verification, AI-based driver behavior monitoring, and cloud-assisted tracking to ensure accurate accountability and misuse prevention. The proposed system adopts these capabilities by combining fingerprint authentication, face-recognition-based driver identification, geo- fencing enforcement, and distance-based KM Lock mechanisms into a unified Python-Raspberry Pi architecture. These additions enable proactive security, allowing the vehicle to intelligently respond to unauthorized users, zone violations, and excessive travel beyond defined thresholds.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2026

DOI: 10.17148/IARJSET.2026.13112

II. METHODOLOGY

The methodology of this project is based on a modular integration of hardware components and Python-driven software logic, allowing the system to operate securely and efficiently within a vehicle environment. The design begins with a structured hardware setup where the Raspberry Pi serves as the central controller, interfacing with authentication modules, GPS sensors, and relay-based control elements. This modular design ensures scalability and simplifies maintenance by isolating functions such as authentication, tracking, and control into independent but interconnected units.

Data acquisition forms the foundation of the system's operation. Driver identity is captured through an RFID or biometric module, while the GPS unit continuously provides real-time coordinates and movement data. The Raspberry Pi simultaneously processes signals related to ignition status, speed, and cumulative distance, ensuring uninterrupted data flow for accurate monitoring. All sensor inputs are handled through appropriate interfaces such as GPIO, UART, or I2C, and Python scripts manage both synchronous and asynchronous data streams, supported by error-handling routines to maintain system reliability even under GPS drift or temporary sensor failures.[3]

The core processing and decision-making logic is executed within the Raspberry Pi. Driver authentication entries are validated against a local or remote database, and movement data is analyzed to determine compliance with predefined rules. The geo-fencing mechanism compares live GPS coordinates with stored perimeter boundaries and initiates alerts or restrictions whenever the vehicle exits the allowed zone. Similarly, the KM Lock functionality continuously computes distance travelled and compares it with the assigned limit; once exceeded, predefined safety actions such as ignition blocking or warning activation are triggered. These processes work together to identify unauthorized drivers, detect misuse, and enforce operational restrictions.[4]

Control actions are executed through relay modules and actuators connected to the Raspberry Pi. Depending on system decisions, the controller can enable or disable ignition circuits, activate buzzers, or display warnings to the driver. All relevant operational data—including driver identity, timestamps, geo-fence status, and KM consumption—is logged locally for future analysis and may optionally be synchronized with a cloud dashboard for remote monitoring.

Finally, the system includes a feedback and testing layer. Real-time information such as authenticated driver details, geographical status, and remaining distance allowance is displayed on an LCD or TFT interface. Each module is individually tested to ensure accurate input–output interaction, and the entire system is evaluated across use cases such as authorized operation, geo-fence breach, and KM limit exceedance. Calibration of GPS accuracy, sensor responsiveness, and authentication reliability ensures that the integrated system performs consistently under real vehicle conditions.

Driver Registration and Face Recognition Pipeline The system implements automated driver registration using a camera module connected to the Raspberry Pi. During registration, 30 grayscale face samples are captured per user using OpenCV's Haar Cascade classifier. These samples from the driver dataset and are used to train the LBPH or KNN recognizer. The trained model is stored locally and loaded during real-time recognition. When the vehicle is accessed, the live video frame is compared against the stored model to verify the driver's identity before ignition activation.

Fingerprint-Based Single-Bit Authentication A fingerprint sensor is integrated with the Raspberry Pi to perform biometric verification. When a fingerprint is matched with a stored template, the system sets a single-bit flag (fpStatus = 1), allowing ignition. If unmatched, the bit remains 0, locking the vehicle. This ensures that only authenticated users can access the controls, providing hardware-level security. Real-Time GPS Parsing and Geo-Fence Enforcement GPS modules provide continuous NMEA data which is parsed using Python's pynmea2 library. The system computes live latitude, longitude, and cumulative distance using the Haversine formula. Geo-fence boundaries are defined dynamically, and crossing these boundaries triggers an immediate motor-cutoff command to the relay via Raspberry Pi.

KM-Lock Distance Computation The system maintains a running log of total distance travelled. When the assigned KM threshold is exceeded, an automated lock signal is generated, forcing the vehicle to stop. This feature is especially useful in rental, corporate, or educational fleet systems.

AI-Based Driver Activity Monitoring To enhance safety, the system periodically checks the driver's face during motion. If the camera detects an unregistered user mid-drive or if the face disappears (possible mismatch or takeover), the system triggers a warning or locks the ignition

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2026

DOI: 10.17148/IARJSET.2026.13112

A. Main Block Diagram

The block diagram below illustrates a comprehensive Raspberry Pi-based smart vehicle security and monitoring system. The system is designed to ensure driver authentication, vehicle safety, motion detection, geo-fencing, and remote notifications using a combination of hardware components and embedded software logic.

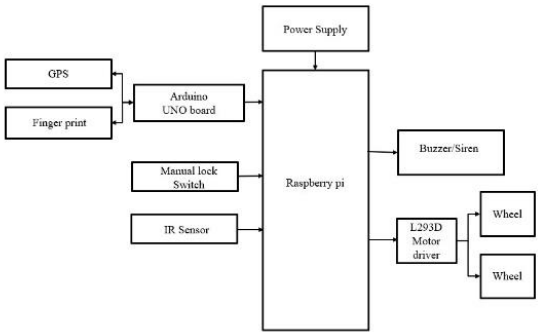

Fig 1. Block Diagram

B. Flowchart Diagram

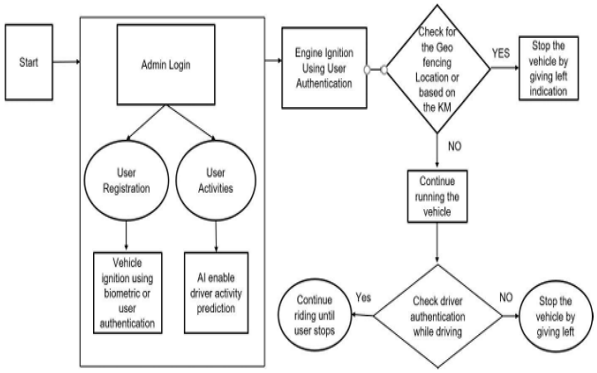

Fig 2. Flow of Implementation

The process flow of the proposed vehicle monitoring and control system, based on the flowchart diagram provided. The system integrates AI-driven predictions, user authentication, and geofencing techniques to enhance vehicle security and operational control is presented.

III. TECHNOLOGY

The technology stack of the proposed system integrates advanced biometric authentication, computer vision, real-time GPS analytics, and embedded control mechanisms to create a secure and intelligent vehicle monitoring framework. At the core of the architecture is the Raspberry Pi, a quad-core ARM-based microcontroller platform running a Linux environment that supports multitasking, high-speed computation, hardware interfacing, and real-time decision making. Its GPIO, UART, I2C, and SPI interfaces enable seamless integration of multiple sensors, biometric modules, and motor control circuits.

To ensure robust identity verification, the system employs a dual-layer authentication approach using both **fingerprint recognition** and **face recognition**. The fingerprint sensor, interfaced through the Adafruit Fingerprint library, performs template generation and matching at the hardware level. Upon successful authentication, it generates a secure **single-bit activation signal** that permits ignition control. In parallel, the camera module supports face detection using OpenCV's Haar Cascade classifier and recognition using LBPH or KNN algorithms. Python-driven model training enables the system to learn unique facial features for each registered driver, enabling highly reliable identity verification even under varying lighting conditions.

GPS technology plays a crucial role in enforcing geo-fencing and distance-based KM Lock functions. The GPS module streams NMEA data to the Raspberry Pi, where the Python library *pynmea2* parses latitude, longitude, speed, and

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2026

DOI: 10.17148/IARJSET.2026.13112

timestamp parameters. Using the Haversine formula, the system computes real-time displacement and cumulative travel distance. These calculations enable automated enforcement actions such as engine lock, alerts, or controlled shutdown whenever the vehicle crosses predefined territorial boundaries or exceeds assigned KM limits.

The Python software stack forms the intelligence layer of the system. Libraries such as OpenCV, NumPy, PySerial, Pynmea2, and Joblib enable image processing, geometric computation, sensor communication, and machine learning model deployment. AI-driven modules perform periodic face re-authentication during vehicle motion to prevent unauthorized driver swapping, ensuring safety and accountability throughout the trip. Additionally, real-time anomaly detection methods help validate sensor data, correct GPS drift, and filter noisy or inconsistent readings.

For hardware-level actuation, relay modules and L293D-based motor drivers' interface with the Raspberry Pi to control ignition, braking, and motor functions. IR sensors connected via GPIO support line detection and assist with controlled motion in prototype environments. Data logging is managed locally using SQLite databases, with optional cloud connectivity via REST APIs or MQTT for remote fleet monitoring, route auditing, and historical activity analysis. Status displays such as I2C LCD or TFT modules present authenticated driver details, geo-fence status, and remaining KM balance in real time.

Together, these technologies establish a multi-layered, AI-assisted, and cyber-physical vehicle security ecosystem. The integration of biometrics, computer vision, GPS intelligence, and embedded actuation enables a scalable, tamper-resistant platform capable of enforcing driver authentication, route compliance, and distance-based operational restrictions with high precision and reliability.

IV. RESULTS AND DISCUSSION

The experimental evaluation of the proposed system was carried out to validate its ability to meet the predefined objectives related to real-time tracking, vehicle control, driver authentication, and secure ignition. The results obtained from prototype testing confirm the successful implementation of all functional modules.
Real-Time GPS Data Extraction and Mapping: The system successfully implemented real-time GPS data extraction using a GPS module interfaced with the Raspberry Pi. Live latitude and longitude values were continuously acquired, processed, and displayed during vehicle operation. In addition to positional data, the system dynamically calculated the cumulative distance traveled by the vehicle. Figure 4.2 illustrates the real-time output showing latitude, longitude, and distance covered, confirming accurate GPS parsing and mapping functionality. Location updates were obtained within a few seconds of initialization, enabling effective live vehicle monitoring.

Motor Control and IR-Based Line Detection System: Motor control was achieved through a motor driver interfaced with the Raspberry Pi, enabling forward motion, directional correction, and controlled stopping of the vehicle. The IR-based line detection system continuously monitored the surface beneath the vehicle and adjusted motor direction accordingly. The prototype vehicle demonstrated stable movement with prompt response to IR sensor inputs, ensuring smooth navigation and reliable path tracking. The successful motion of the developed bot is shown in Fig. 4.1, validating effective motor and sensor integration.

Driver Registration and Identification: The system effectively implemented driver registration and identification using biometric authentication. Authorized drivers were enrolled into the system database, and access was granted only after successful verification. During testing, authentication decisions were generated within a short time interval, preventing unauthorized vehicle access. Driver identity and session details were recorded during operation, ensuring accountability and traceability of vehicle usage.

Single-Bit Authentication for Vehicle Activation and Activity Recording: A single-bit authentication mechanism was integrated to control vehicle ignition. Upon successful driver verification, a binary activation signal enabled vehicle movement, whereas failed authentication kept the system in a locked state. This mechanism ensured secure ignition control while simplifying system logic. Driver activity, including authentication status and operational duration, was recorded for each session, confirming the effectiveness of the single-bit control approach in enforcing secure vehicle operation.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2026

DOI: 10.17148/IARJSET.2026.13112

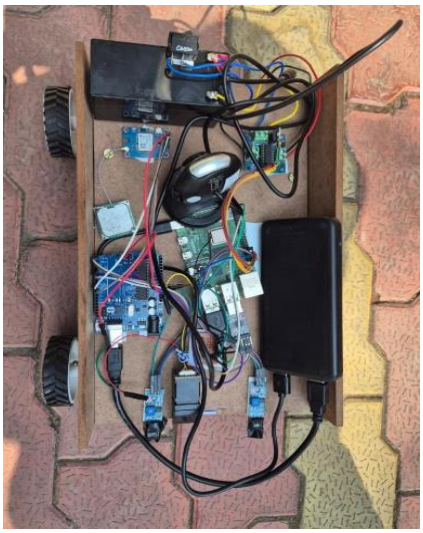

Fig 4.1 Prototype vehicle (bot)

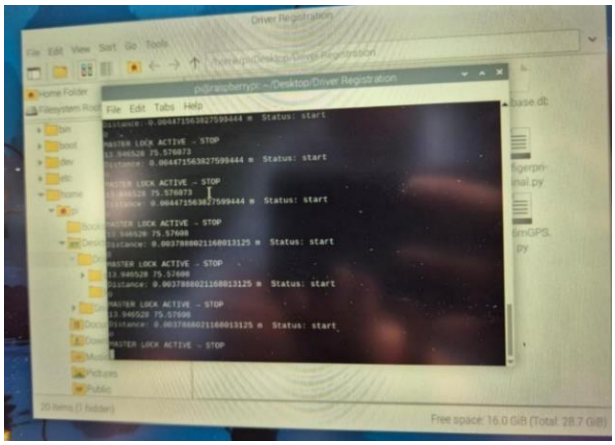

Fig. 4.2 Real-time GPS output showing latitude, longitude, and distance covered by the vehicle

V.CONCLUSION

The implemented system delivers an advanced, multi-layered solution for secure and intelligent vehicle monitoring by integrating biometric authentication, computer-vision-based driver identification, real-time GPS analytics, geofencing, and distance-based KM Lock enforcement. The combination of fingerprint verification, face-recognition models, and single-bit ignition control ensures that only authenticated users can operate the vehicle, significantly reducing risks of unauthorized access, misuse, and theft. Continuous GPS tracking and automated geo-fence monitoring enable the system to respond instantly to boundary violations, while KM-based travel restrictions provide precise operational control for fleet management and rental applications.

The inclusion of AI-assisted driver re-authentication during vehicle movement enhances safety by preventing driver swapping or unauthorized takeover. Embedded Python logic, supported by a Raspberry Pi computing platform, ensures reliable decision-making and real-time actuation of relays, motor drivers, and alert mechanisms. Test results confirm that the system performs consistently across core modules—authentication, tracking, movement control, and boundary enforcement—demonstrating its practicality for modern intelligent transport systems.

Overall, the project establishes a robust foundation for next-generation vehicle security architectures. Future enhancements such as GSM/GPRS alerts, cloud dashboards, mobile app integration, onboard CNN-based behaviour detection, and predictive analytics can further elevate its scalability, remote accessibility, and intelligence, making it highly suitable for real-world deployment in commercial fleets, institutional transport, and high-security automotive environments.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 1, January 2026

DOI: 10.17148/IARJSET.2026.13112

ACKNOWLEDGEMENTS

We express our heartfelt gratitude to JNN College of Engineering, Shimoga, for providing the academic environment, laboratory facilities, and encouragement that enabled us to successfully carry out this project. We are deeply indebted to our guide, **Mrs. Ujwala B. S**., for her continuous support, expert guidance, and thoughtful suggestions throughout the development of **"Driver Identification and Activity Tracking with Geo-Fencing & Number KM Lock Using Python and Raspberry Pi."** Her mentorship played a vital role in shaping our ideas and overcoming the technical challenges encountered during the project.

We also extend our sincere thanks to the faculty members of the Department of Electronics and Communication Engineering for their valuable feedback and motivation at different stages of our work, which greatly helped in refining the overall outcome of the project.

We would like to express our profound gratitude to our parents for their financial support, encouragement, and unwavering belief in our efforts. Their constant motivation and assistance made it possible for us to procure the necessary components and complete the project successfully.

Finally, we acknowledge the support and cooperation of our peers and friends, who contributed directly or indirectly to the smooth progress of this work.

REFERENCES

[1]. Upendra, Modem, and Varun Reddy. "Smart GPS Based Vehicle Speed Limit Controller on zone identification using Geo-Fencing Algorithm." In 2024 3rd International Conference for Innovation in Technology (INOCON), pp. 1-6. IEEE, 2024.

[2]. Zhao, Dongyang, Yuqing Chen, and Shuanghe Yu. "Tracking and speed estimation of ground vehicles using aerial-view videos." In 2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE), pp. 597-601. IEEE, 2020.

[3]. Nel, Francois, and Mkhuseli Ngxande. "Driver activity recognition through deep learning." In 2021 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA), pp. 1-6. IEEE, 2021.

[4]. Karim, Danish, and Jaspal Singh. "Development of automatic geofencing and accidental monitoring system based on GPS technology." International Journal of Computer Science, Engineering and Applications 3, no. 4 (2013): 57

[5]. Rathore, Aditya Singh. "Lane detection for autonomous vehicles using OpenCV library." International Research Journal of Engineering and Technology 6, no. 1 (2019): 1326-1332.

[6]. Khan, Zain."Real-Time Vehicle Tracking System Using Arduino and GPS." Hackster.io.