International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 05 May-2026 | Impact Factor: 3.5

Smart Vehicle Safety System Using IoT and Machine Learning for Accident Detection and Adaptive Cruise Control

Akash M, Mahibalan V, Ranjanilakshmi P
Dept. of Electrical and Electronics Engineering
Coimbatore Institute of Engineering and Technology
Coimbatore, Tamil Nadu, India
akashm003@ciet.edu.in | mahibalanv008@ciet.edu.in

Supervisor: **Ms. R. Karthigayini, M.E.**
Dept. of Electrical and Electronics Engineering
Coimbatore Institute of Engineering and Technology
Coimbatore, Tamil Nadu, India
ranjanilakshmip011@ciet.edu.in

How to Cite this Article:
P, R., V, M. & M, A. (2026). Smart Vehicle Safety System Using IoT and Machine Learning for Accident Detection and Adaptive Cruise Control. International Journal of Creative and Open Research in Engineering and Management, 02(05). https://doi.org/10.55041/ijcope.v2i5.043

https://doi.org/10.55041/ijcope.v2i5.043

Abstract—Road accidents remain a leading cause of global fatalities, with over 1.35 million deaths annually according to WHO. Existing vehicle safety systems rely on passive mechanisms—airbags, ABS, ESC—that mitigate injury severity but cannot predict or prevent accidents, nor communicate emergencies automatically. This paper presents a Smart Vehicle Safety System that integrates Internet of Things (IoT) sensing with embedded Machine Learning (TinyML) to address these limitations. The system employs an ESP32 microcontroller interfaced with an MPU6050 accelerometer, HC-SR04 ultrasonic sensor, NEO-6M GPS module, SIM800L GSM module, NRF24L01 wireless transceiver, and a DC motor with PWM driver. A Random Forest classifier, trained on accelerometer data and compressed into a C++ model header via Eloquent ML, performs real-time on-device accident detection without cloud dependency. Upon detection, the GPS coordinates are transmitted as SMS alerts to emergency contacts through the GSM module, while accident records—including timestamp, location, and impact intensity—are uploaded to a Firebase Realtime Database for remote monitoring. A zone-based adaptive cruise control feature uses RF signals from roadside NRF transmitters to automatically regulate vehicle speed in critical zones such as school areas, cross-roads, and city limits. Experimental validation demonstrated 94.3% accident detection accuracy across 35 simulated impact events, emergency SMS delivery within 8–12 seconds, and reliable zone-based speed regulation with sub-200 ms response. The system provides a cost-effective, scalable solution for next-generation intelligent transportation systems.

Keywords—IoT; TinyML; ESP32; Accident Detection; Adaptive Cruise Control; GPS; GSM; Firebase; NRF24L01; Random Forest; Vehicle Safety System

I. Introduction

Road traffic accidents constitute a critical public health crisis worldwide. The World Health Organization (WHO) estimates that 1.35 million people die each year due to road accidents, with an additional 20–50 million suffering non-fatal injuries. In India alone, over 1.5 lakh fatalities are recorded annually, making it one of the worst-affected nations. The primary causative factors include human error, driver fatigue, delayed emergency response, and inadequate real-time situational awareness.

Conventional vehicle safety systems—airbags, Anti-lock Braking Systems (ABS), and Electronic Stability Control (ESC)—are reactive; they operate only after an accident has occurred and provide no means of automatic emergency communication. Advanced driver assistance systems available in premium vehicles are expensive, proprietary, and rarely integrated with emergency communication networks in developing countries.

The emergence of low-cost microcontrollers such as the ESP32, combined with miniaturized sensors and cloud

3) NEO-6M GPS Module:

The u-blox NEO-6M GPS module communicates with the ESP32 via UART (RX: GPIO16, TX: GPIO17) at 9600 baud. It provides latitude and longitude coordinates with accuracy up to 2.5 m CEP. The TinyGPS++ library parses NMEA sentences to extract location data. GPS fix is maintained continuously so that coordinates are immediately available when an accident is detected, minimising the delay between detection and SMS dispatch.

4) SIM800L GSM Module:

The SIM800L quad-band GSM/GPRS module is interfaced via a dedicated UART port (TX: GPIO32, RX: GPIO33). It is initialized using AT commands (AT, AT+CMGF=1 for SMS text mode) and transmits emergency SMS messages containing accident location as a Google Maps hyperlink (https://maps.google.com/?q=lat,lng) to pre-programmed emergency contacts. Additionally, it

connectivity, has created opportunities to develop intelligent vehicle safety systems that are both affordable and effective. Furthermore, recent advances in embedded Machine Learning—often termed TinyML—enable deployment of inference models directly on microcontrollers, eliminating reliance on cloud connectivity for time-critical decisions such as accident detection.

This paper presents a comprehensive Smart Vehicle Safety System with two primary functional modules: (1) an accident detection and emergency alert module that uses a TinyML Random Forest model to classify accelerometer data and triggers GPS-tagged SMS alerts via GSM upon detection, and (2) an adaptive cruise control (ACC) module that uses NRF24L01 zone transmitters to automatically regulate vehicle speed in predefined critical areas. The system is validated through hardware prototype testing.

II. Literature Review

Kumar and Singh [1] developed an IoT-based accident detection system using accelerometers, vibration sensors, and GPS integrated over cellular networks. Their study demonstrated that automated emergency notification reduces average response time by 40% compared to manual reporting, though false-trigger rates of up to 15% were reported in high-vibration environments such as unpaved roads.

Sharma and Mehta [2] conducted a comparative evaluation of ML algorithms—Support Vector Machines (SVM), Decision Trees, and Artificial Neural Networks (ANN)—for accident prediction using vehicle speed, braking patterns, and road condition inputs. They concluded that ensemble methods outperform single-model approaches, achieving accuracy exceeding 91%, but noted the need for large labelled datasets and high computational resources.

Lee and Kim [3] advanced ACC systems by incorporating ML-based dynamic speed adjustment using radar and LiDAR inputs. Their model reduced rear-end collision incidents by 28% in simulation but highlighted cost and sensor-fusion complexity as barriers to mass adoption.

Patel and Verma [4] proposed a cloud-integrated IoT framework where sensor data is streamed to ML models hosted on cloud platforms for hazard detection. While achieving high accuracy, cloud-dependent inference introduced latencies of 200–800 ms, which is unacceptable for real-time accident response. The present work addresses this gap through on-device TinyML inference.

Gupta and Rao [5] demonstrated that combining IoT with cloud computing for post-accident management enables location data to reach emergency services in under 30 seconds, substantially improving survival rates. Nalayini et al. [6] further explored RF-based speed deterrence in school zones, which forms the conceptual basis of the ACC module presented in this work.

III. System Architecture

The proposed system comprises two hardware units that operate in tandem: a Transmitter Unit for zone-based speed signaling and a Vehicle Unit for integrated sensing, processing, communication, and actuation. Fig. 1 illustrates the overall system architecture.

A. Transmitter Unit

The Transmitter Unit is a fixed roadside device installed at critical zones such as school areas, cross-roads, and city boundaries. It consists of an Arduino Uno microcontroller, an supports HTTP GET requests for Firebase data upload over GPRS when Wi-Fi is unavailable.

5) NRF24L01+ Transceiver:

The NRF24L01+ module operates on the 2.4 GHz ISM band and is interfaced with the ESP32 via SPI (CE: GPIO2, CSN: GPIO5). The RF24Network library manages node addressing (master node: 00, transmitter node: 01). The receiver continuously listens for zone data packets from roadside transmitters and forwards decoded zone codes to the ACC control logic.

6) DC Motor and PWM Driver:

A 12V DC motor represents the vehicle's traction system. The ESP32 drives the motor via a BJT-based PWM driver circuit (BC337 and BD139 transistors) using the LEDC PWM peripheral on GPIO14 at 500 Hz, 8-bit resolution. The duty cycle (0–255) is mapped to effective motor speed, enabling smooth and precise ACC regulation.

IV. Methodology

A. TinyML Accident Detection Model

A Random Forest classifier was trained offline using a labelled dataset of 1,200 accelerometer readings collected from two classes: normal driving events (sharp turns, speed bumps, hard braking) and accident events (frontal impact, side impact, rollover simulations). Two features were extracted per sample: the resultant acceleration magnitude ($\sqrt{(ax^2+ay^2+az^2)}$) and the rate-of-change of resultant magnitude between consecutive samples.

The model was trained in Python using scikit-learn (100 estimators, max depth 8, min samples split 4) and achieved 96.2% cross-validated accuracy on the training set. The trained model was exported to a C++ header file (modelav.h) using the Eloquent ML library and compiled into the ESP32 firmware. The exported model occupies approximately 14 KB of flash memory, well within the ESP32's 4 MB capacity.

During runtime, accelerometer data is sampled at 50 Hz. Every 100 ms, the two feature values are computed and passed to the classifier. If the classifier returns an accident label AND the ultrasonic sensor registers a distance greater than 50 cm (ruling out stationary obstacle impact), the event is confirmed as an accident. This dual-condition trigger reduces false-positive rate to under 5%.

B. Emergency Response Pipeline

Upon accident confirmation, the following automated response sequence is executed: (1) The GPS module provides current latitude and longitude. (2) The buzzer sounds a continuous alert tone. (3) The LCD displays ACCIDENT DETECTED with GPS coordinates. (4) The GSM module transmits an SMS to emergency contacts: Accident Alert! Location: https://maps.google.com/?q=lat,lng Vehicle: TN99AS8888. (5) An HTTP GET request uploads accident data—timestamp, vehicle ID, GPS coordinates, impact intensity—to the Firebase Realtime Database. (6) Firebase Cloud Messaging

NRF24L01+ 2.4 GHz wireless transceiver, three toggle switches for manual zone selection, and a 12V/1A DC power adapter.

When a zone switch is activated, the Arduino encodes the corresponding zone code (1=School Zone, 2=Cross Road, 3=City Zone, 0=Normal) and continuously broadcasts it at 1-second intervals via the NRF24L01 module operating on channel 90 at 2 Mbps data rate. The RF signal propagates over a coverage radius of approximately 10–15 meters, sufficient for typical zone entry points.

B. Vehicle Unit

The Vehicle Unit is the central intelligent module mounted on the vehicle. It is built around the ESP32-WROOM-32 microcontroller, which provides dual-core 240 MHz processing, built-in Wi-Fi, Bluetooth, and extensive GPIO connectivity. The following subsections describe each component.

1) MPU6050 Accelerometer / Gyroscope:

The GY-521 MPU6050 module provides 3-axis acceleration (±8g range) and 3-axis gyroscopic data via I2C communication (SDA: GPIO21, SCL: GPIO22). Sampled at 50 Hz, it captures sudden impacts, rapid deceleration, and vehicle rollover—key signatures of accident events. The raw acceleration values are fed as input features to the embedded TinyML model.

2) HC-SR04 Ultrasonic Sensor:

The ultrasonic sensor measures the distance to the nearest forward obstacle using the time-of-flight principle. A 10 µs trigger pulse is emitted at 40 kHz; the echo pulse duration is measured and converted to distance using: Distance (cm) = (pulse duration in µs × 0.034) / 2. The sensor provides measurements in the range 2–400 cm with ±3 mm accuracy, enabling collision avoidance and distinguishing genuine accidents from braking-induced false triggers.

TABLE I. ZONE-BASED SPEED CONTROL PARAMETERS

Zone Code	Zone Type	Speed Limit	PWM Duty
1	School Zone	20 km/h	45%
2	Cross Road	30 km/h	60%
3	City Zone	40 km/h	75%
0	Normal Road	Unrestricted	100%

B. Emergency Communication Latency

Emergency SMS delivery was evaluated over 10 accident trigger events under a 4G cellular network. The average end-to-end latency from accident detection to SMS receipt was 9.4 seconds (min: 7.8 s, max: 12.1 s). The primary latency component was GSM module initialization and SMS transmission (~6 s). GPS fix latency (cold start: 35–40 s; warm start: 1–3 s) is managed by maintaining a continuous GPS fix during vehicle operation, contributing negligibly (<0.5 s) to emergency response time when the vehicle is already in motion.

Firebase upload latency averaged 2.3 seconds over Wi-Fi and 4.8 seconds over GPRS, confirming (FCM) dispatches push notifications to registered mobile app instances.

C. Adaptive Cruise Control Logic

When the NRF receiver detects a valid zone packet, the ESP32 reads the zone code and applies the corresponding speed limit policy. The DC motor PWM duty cycle is adjusted as per Table I. When the zone signal is absent (code 0), the vehicle resumes normal potentiometer-controlled speed. A brief buzzer beep confirms each zone transition to the driver.

D. Firebase Cloud Integration

The ESP32 connects to a configured Wi-Fi access point on startup. Accident records are pushed to the Firebase Realtime Database as JSON objects via HTTP GET requests to a PHP relay script hosted at the project server. The Firebase console provides a tabular web dashboard displaying date, vehicle registration number, accident location (as a clickable Google Maps link), and clearance status. Firebase Cloud Messaging delivers real-time push notifications to traffic control personnel and emergency responders.

V. Results and Discussion

A. Accident Detection Performance

Hardware testing was conducted on a bench-top prototype with the MPU6050 mounted on a rigid board. Thirty-five test events were simulated across three impact categories: frontal impact (n=15), lateral impact (n=12), and rollover simulation (n=8). The Random Forest classifier correctly identified 33 of 35 events, yielding a detection accuracy of 94.3%. Two false negatives occurred in low-intensity lateral impacts near the decision boundary. Zero false positives were recorded during 20 normal driving manoeuvre tests (sharp braking, speed bumps at 40 km/h).

VII. Conclusion

This paper presented a fully integrated Smart Vehicle Safety System combining IoT-based sensing, embedded TinyML inference, cellular emergency communication, and RF-based adaptive cruise control on a low-cost ESP32 platform. The key contributions of this work are:

1) Deployment of a compressed Random Forest accident classifier directly on the ESP32 microcontroller, enabling real-time, cloud-independent accident detection with 94.3% accuracy.

2) An automated GPS-tagged emergency SMS pipeline achieving average end-to-end latency of 9.4

real-time data availability on the cloud dashboard. FCM push notifications were received within 3–5 seconds of cloud upload across three test mobile devices.

C. Adaptive Cruise Control Performance

Zone detection was tested across 25 vehicle pass-through trials at three transmitter locations. The nRF receiver successfully captured zone codes in 24 of 25 trials (96% reception success rate). The single missed detection occurred due to physical obstruction reducing RF signal strength. Speed adjustment was executed within 180 ms of zone code reception. The buzzer confirmation tone was produced in all successful detections.

Motor speed regulation accuracy was measured using an optical tachometer. The target PWM duty cycles (Table I) produced motor speeds within ±4% of the intended speed setpoints, confirming adequate precision for practical ACC implementation.

D. Obstacle Detection

The HC-SR04 sensor was evaluated for obstacle detection reliability at distances ranging from 5 cm to 200 cm. Measured distances deviated by less than ±3 cm from ground truth across all trials. The 50 cm threshold trigger for buzzer activation was consistently invoked at the correct distance. When obstacle distance fell below 50 cm, the system displayed "OBSTACLE DETECT" on the LCD and activated the buzzer within 20 ms, providing timely driver warning.

VI. Comparative Analysis

Table II compares the proposed system against representative existing approaches from the literature, highlighting the advancement offered by on-device TinyML inference and the combined ACC + emergency alert functionality.

TABLE II. COMPARISON WITH EXISTING SYSTEMS

Feature	Kumar [1]	Sharma [2]	Lee [3]	Proposed
On-device ML	No	No	Partial	Yes
Cloud Dependency	Yes	Yes	No	Optional
Auto SMS Alert	Yes	No	No	Yes
ACC Feature	No	No	Yes	Yes
Detection Acc.	~85%	91%	N/A	94.3%
SMS Latency	~15s	N/A	N/A	9.4s

seconds, significantly faster than manual emergency reporting.

3) A zone-based adaptive cruise control mechanism using NRF24L01 RF transceivers that reliably regulates vehicle speed in school zones, cross-roads, and city limits with 96% zone detection success rate.

4) Real-time cloud monitoring and push notification via Firebase Realtime Database and FCM, enabling remote oversight by traffic authorities and emergency responders.

The prototype demonstrates that cost-effective, intelligent vehicle safety systems are feasible for deployment in developing-country contexts where premium ADAS solutions are inaccessible. The modular architecture supports future extensions including vehicle-to-vehicle (V2V) communication, camera-based driver drowsiness detection using CNNs, integration with national emergency dispatch APIs (e.g., Dial 112), and enhanced TinyML models trained on real-world accident datasets to improve detection accuracy and robustness across diverse road conditions.

Acknowledgment

The authors sincerely thank Dr. S. Gokul, M.E., Ph.D., Head of Department, and Ms. R. Karthigayini, M.E., Assistant Professor, Department of Electrical and Electronics Engineering, Coimbatore Institute of Engineering and Technology, for their invaluable guidance and continuous support throughout this project. The authors also acknowledge the management of CIET for providing laboratory facilities and resources essential for the completion of this work.

References

[1] P. Kumar and R. Singh, "IoT-based accident detection and emergency notification system," IEEE Access, vol. 11, 2023.

[2] A. Sharma and D. Mehta, "Machine learning approaches for road accident prediction," J. Intelligent Transportation Systems, vol. 25, no. 4, 2021.

[3] J. Lee and H. Kim, "Intelligent adaptive cruise control using machine learning techniques," IEEE Trans. Vehicular Technology, vol. 71, no. 3, 2022.

[4] S. Patel and K. Verma, "Smart vehicle systems using IoT and machine learning integration," Int. J. Smart Transportation, vol. 8, 2022.

[5] N. Gupta and S. Rao, "Real-time accident detection and emergency response system using IoT," IEEE Internet of Things Journal, vol. 7, no. 10, 2020.

[6] C. M. Nalayini, P. Sreemathi, and B. Nanditha, "Deterrence of Accident Using IoT," J. Trends in Computer Science and Smart Technology, vol. 4, no. 2, 2022.

[7] N. Pathik et al., "AI Enabled Accident Detection and Alert System Using IoT and Deep Learning for Smart Cities," Sustainability, vol. 14, no. 13, 2022.

[8] S. Oza and S. Rathod, "Object Detection using IoT and ML to Avoid Accident and Improve Road Safety," IJERT, vol. 9, no. 6, 2020.

[9] Espressif Systems, "ESP32 Technical Reference Manual," [Online]. Available: https://www.espressif.com

[10] Google Firebase, "Firebase Realtime Database Docs," [Online]. Available: https://firebase.google.com