International Journal of Electronics and Communication Engineering (IJECE)
ISSN(P): 2278-9901; ISSN(E): 2278-991X
Vol. 15, Issue 1, Jan–Jun 2026; 1–10

International Academy of Science, Engineering and Technology
IASET
Connecting Researchers; Nurturing Innovations

IOT-DRIVEN VEHICLE CRASH DETECTION AND EMERGENCY RESPONSE SYSTEM

Parasakthi. B[1], Rajeswari. P[2] & A. Yogeshwaran[3]

[1]PG Students, Electronics and Communication Engineering, Dhanalakshmi Srinivasan Engineering College(Autonomous), Perambalur, India

[2,3]Associate Professor, Electronics and Communication Engineering, Dhanalakshmi Srinivasan Engineering College (Autonomous), Perambalur, India

ABSTRACT

Rapid detection of vehicular accidents and timely notification of emergency services are critical factors in reducing fatalities and mitigating the severity of injuries. This paper presents an IoT-driven Vehicle Crash Detection and Emergency Response System based on the ESP32 microcontroller platform. The proposed system integrates an Vibration sensor to detect sudden impacts or abnormal vehicle movements, indicative of a collision. Upon detection, the ESP32 processes the data in real-time and triggers an automated alert mechanism, sending critical information such as vehicle location and impact severity to predefined contacts and emergency responders via GSM, Wi-Fi, or MQTT protocols. The system also includes optional components such as GPS modules for accurate geolocation and cloud-based monitoring for real-time tracking. Experimental results demonstrate that the system can reliably detect accidents, minimize response time, and potentially save lives by enabling swift emergency intervention. The modular design ensures scalability and adaptability for various vehicle types, paving the way for smarter, safer transportation systems.

KEYWORDS: *IoT, ESP32, Vehicle Accident Detection, Emergency Alert System, Real-Time Monitoring, Vibration, GPS, Smart Transportation*

Article History

Received: ***10 Apr 2026*** **|** ***Revised:*** ***15 Apr 2026*** **|** ***Accepted:*** ***17 Apr 2026***

INTRODUCTION

Road traffic accidents have become one of the leading causes of death and severe injuries worldwide. According to recent statistics, millions of people are injured or killed annually due to vehicular collisions, and a significant proportion of fatalities occur due to delayed emergency response. Rapid detection of accidents and timely alerting of emergency services is therefore a critical requirement in modern intelligent transportation systems. Traditional methods for accident detection rely primarily on manual reporting or centralized traffic monitoring systems, which often suffer from delayed response times, inaccurate reporting, and limited real-time monitoring capabilities.

Advances in Internet of Things (IoT) technology have created opportunities for more effective and automated solutions for vehicle safety and accident management. IoT enables vehicles to be equipped with smart sensors and connected microcontrollers that continuously monitor driving conditions and detect abnormal events in real time. In this context, the ESP32 microcontroller has emerged as a versatile and cost-effective platform for embedded IoT applications.

With its integrated Wi-Fi and Bluetooth capabilities, high processing power, and low energy consumption, ESP32 allows for seamless communication between the vehicle and emergency response systems, facilitating immediate action when accidents occur.

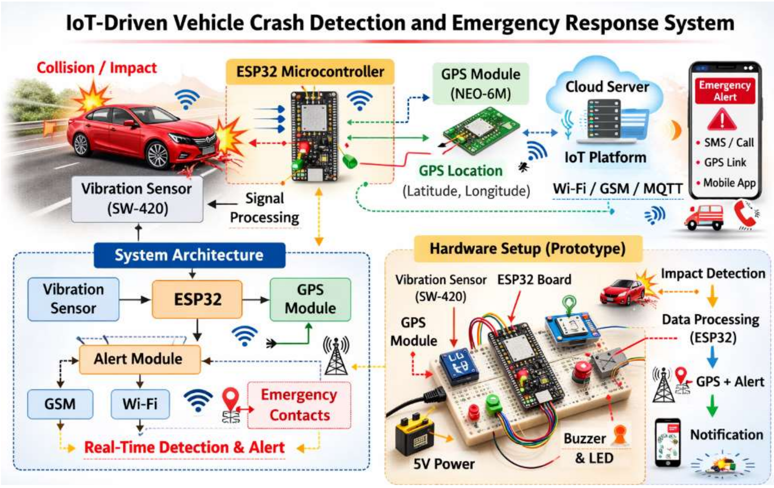

Figure 1: Vehicle Crash Systems.

The primary goal of an IoT-driven vehicle crash detection system is to identify sudden vehicle impacts, collisions, or abnormal motion patterns indicative of an accident and automatically alert emergency contacts, medical services, or law enforcement. Instead of complex accelerometer and gyroscope modules, the proposed system employs a **vibration sensor (such as SW-420)** to detect sudden shocks and impacts. The vibration sensor outputs a digital HIGH signal when it experiences strong mechanical vibrations, allowing the ESP32 to recognize potential collisions. This approach simplifies sensor integration while maintaining reliable accident detection.

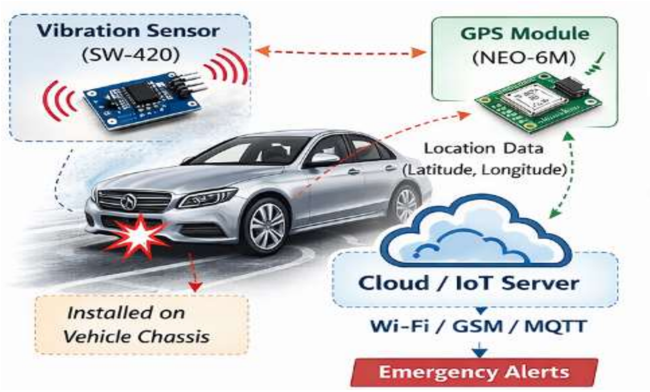

Figure 2: The Detection of a Crash using Sensors.

Upon detection of a crash, the ESP32 processes the sensor signal in real time and triggers a communication protocol to transmit critical information. These alerts typically include the vehicle's GPS coordinates, the estimated severity of the impact, and other relevant data that enable first responders to reach the accident site quickly. This automated approach significantly reduces response time compared to traditional methods, increasing the chances of survival and reducing the severity of injuries for accident victims.

Moreover, IoT-based crash detection systems can be integrated into smart city infrastructure, allowing vehicles to communicate with cloud platforms or centralized monitoring centers. Such integration enables real-time data collection, analytics, and visualization, which can further assist traffic authorities in identifying accident-prone areas, monitoring accident trends, and implementing preventive measures. In addition to safety, these systems also enhance vehicle insurance management by providing reliable data about collisions, reducing fraudulent claims and simplifying the claim verification process.

Despite the benefits, designing an effective accident detection system poses several challenges. Accurate detection requires proper placement and threshold calibration of the vibration sensor to avoid false positives from abrupt braking, potholes, or uneven road surfaces. Real-time data processing must be efficient to ensure instant alerts without overloading the system.

PROPOSED SYSTEM

The proposed IoT-Driven Vehicle Crash Detection and Emergency Response Systemleverages the ESP32 microcontroller and a vibration sensor to overcome these challenges. By integrating real-time sensing, intelligent data processing, and automated alerting, the system provides a proactive approach to vehicle safety.

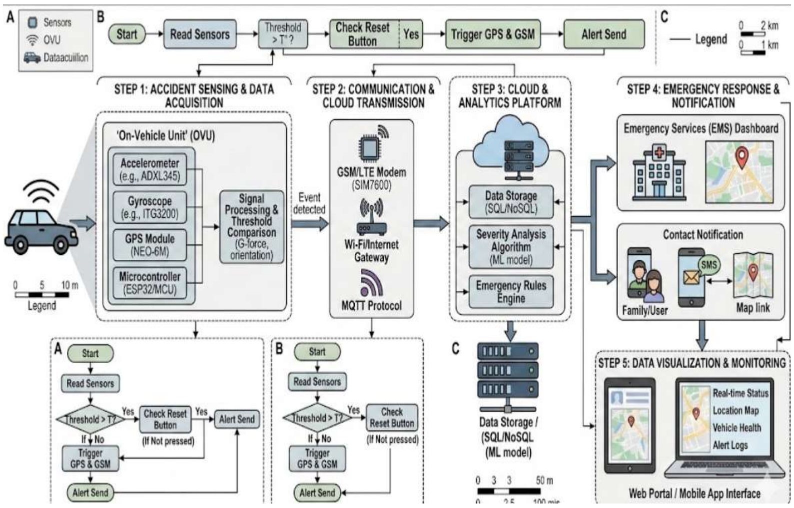

Figure 3: Over all Architecture.

Experimental implementation shows that the system can accurately detect collisions, promptly notify emergency contacts, and improve overall road safety. The modular design and IoT connectivity ensure that the system is adaptable for future upgrades, including cloud analytics, predictive accident prevention, and integration with vehicle-to-everything (V2X) communication networks. The need for real-time accident detection and emergency response has become critical in modern transportation. IoT-enabled solutions provide an effective way to monitor vehicle conditions, detect collisions, and communicate critical information to emergency services. The ESP32-based crash detection system combines sensor technologies, wireless communication, and automated alerting to improve response times, enhance road safety, and reduce accident-related fatalities. The subsequent sections of this paper present the detailed system architecture, sensor integration, communication protocols, and experimental results, highlighting the potential of IoT-driven solutions in transforming vehicular safety.

WORKING PRINCIPLE

The IoT-Driven Vehicle Crash Detection and Emergency Response Systemoperates by continuously monitoring vehicular vibrations and abnormal events and automatically sending alerts to emergency responders upon detecting a collision. The working principle of the system can be explained in the following steps:

Step1: Vibration Detection

The system employs a vibration sensor (SW-420) mounted on the vehicle chassis to monitor mechanical shocks or impacts. During normal driving conditions, the sensor remains in an inactive state (digital LOW). In the event of a collision, sudden mechanical vibrations or shocks cause the sensor output to transition to a digital HIGH state. This signal serves as an initial indicator of a potential accident.

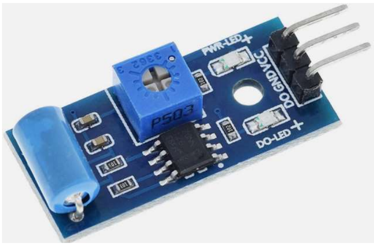

Figure 4: Vibration Sensor.

Step 2: Signal Processing

The digital signal from the vibration sensor is sent to the ESP32 microcontroller, which acts as the central processing unit. The microcontroller continuously monitors the sensor input to determine whether the vibration exceeds a predefined threshold. To prevent false positives caused by minor bumps, potholes, or braking events, the system implements a debouncing and threshold-check algorithm, ensuring that only significant shocks trigger further action.

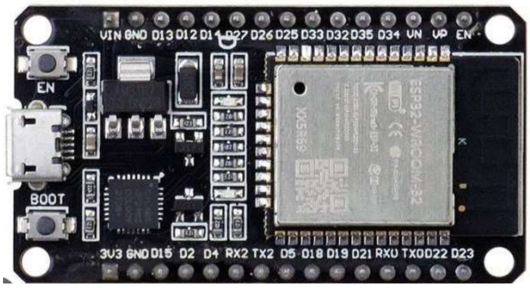

Figure 5: ESP32 Controller.

Step 3: GPS Data Acquisition

Upon confirming a potential collision, the ESP32 interfaces with a GPS module to acquire the vehicle's precise location coordinates (latitude and longitude). Accurate geolocation information is essential for emergency responders to reach the accident site promptly. The GPS data is retrieved in real time and prepared for transmission alongside the impact alert.

Step 4: Communication and Alert Transmission

Once the crash is detected and GPS coordinates are acquired, the ESP32 initiates the alert communication module. Depending on the system configuration, alerts can be sent via Wi-Fi, GSM, or MQTT protocols to predefined emergency contacts, cloud servers, or centralized monitoring systems. The alert message typically contains the vehicle ID, accident timestamp, location coordinates, and a brief description of the detected impact.

Figure 6: GPS Module.

Step 5: Notification to Emergency Responders

The transmitted alert is received by emergency contacts or monitoring centres, which can include medical personnel, law enforcement, or family members. This enables a rapid response, potentially reducing the severity of injuries and saving lives. If integrated with a cloud-based monitoring platform, the system can also provide real-time visualization and analytics for traffic authorities and fleet managers.

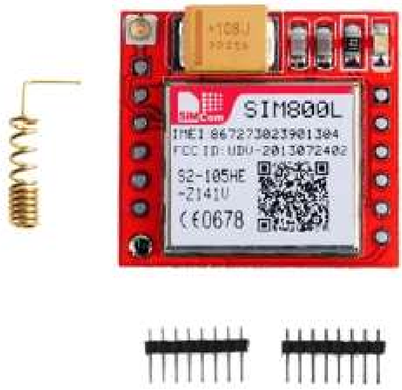

Figure 7: GPS Module.

Step 6: System Reset and Continuous Monitoring

After sending the alert, the system automatically resets to its monitoring state, ensuring continuous surveillance of the vehicle. This allows the system to detect subsequent incidents without manual intervention, maintaining a proactive approach to vehicle safety.

HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements

The proposed system is designed with **cost-effective and readily available components** to ensure reliable detection of vehicle collisions and timely alerts. The central processing unit of the system is the **ESP32 microcontroller**. Equipped with built-in Wi-Fi and Bluetooth, the ESP32 handles real-time processing of sensor data and communicates with emergency responders or cloud servers. Its high processing speed, low power consumption, and integrated wireless capabilities make it ideal for IoT-based accident detection systems.

The primary sensing element is a **vibration sensor (SW-420)**, which is used to detect sudden shocks or collisions. The sensor produces a digital HIGH signal when a significant vibration occurs, signaling a potential accident to the ESP32. This choice of sensor reduces system complexity while maintaining high reliability, as it can filter minor disturbances like bumps or potholes.

To provide accurate geolocation, the system integrates a **GPS module** such as the NEO-6M. The GPS module continuously receives satellite signals and calculates the vehicle's latitude and longitude. In the event of a detected collision, these coordinates are included in the alert message sent to emergency contacts, enabling responders to locate the accident site quickly.

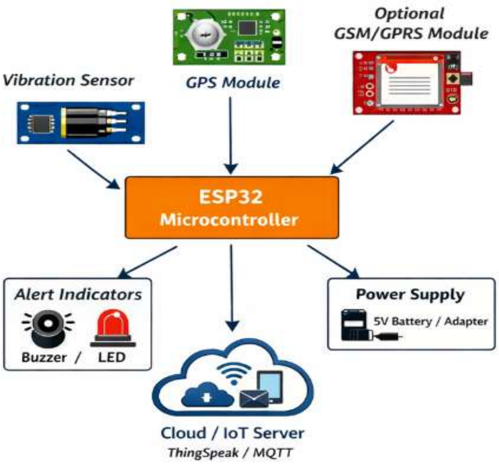

Figure 8: Block Diagram of AVAS.

For transmitting alerts via cellular networks, a **GSM/GPRS module (e.g., SIM800L)** is included as an optional component. This module allows the ESP32 to send SMS alerts to predefined contacts, ensuring that emergency notifications are delivered even in areas without Wi-Fi connectivity.

To provide immediate local feedback, the system uses **relays, LEDs, and buzzers**. These components alert the driver or nearby individuals that an accident has been detected, which can be useful for manual intervention or additional safety measures.

Finally, a **stable 5V power supply** is required to ensure continuous operation of the ESP32 and all connected modules. Power can be supplied using a vehicle battery adapter or a regulated battery system. Supporting hardware, such as connecting wires and a breadboard or PCB, is also necessary for assembling the modules securely and reliably.

Software Requirements

The system's functionality depends on firmware programming, communication protocols, and IoT tools to process sensor data and transmit alerts. **Arduino IDE** is the primary development environment used to write, compile, and upload the ESP32 firmware. It provides an intuitive interface for programming the microcontroller and debugging the system.

The **ESP32 board package** must be installed within the Arduino IDE to enable support for ESP32 development. Additionally, libraries such as **WiFi, HTTPClient, and TinyGPS++** are required for handling Wi-Fi connectivity, parsing GPS data, and transmitting data to cloud servers.

For IoT communication, the system uses **MQTT or HTTP protocols**, implemented through libraries like PubSub Client or HTTP Client. These protocols allow real-time transmission of collision alerts and GPS data to monitoring platforms or mobile devices.

Integration with a **cloud or IoT platform**, such as ThingSpeak, Blynk, or a custom MQTT broker, enables centralized monitoring and visualization of accident events. This allows emergency responders and fleet managers to track incidents, analyze data trends, and implement preventive measures.

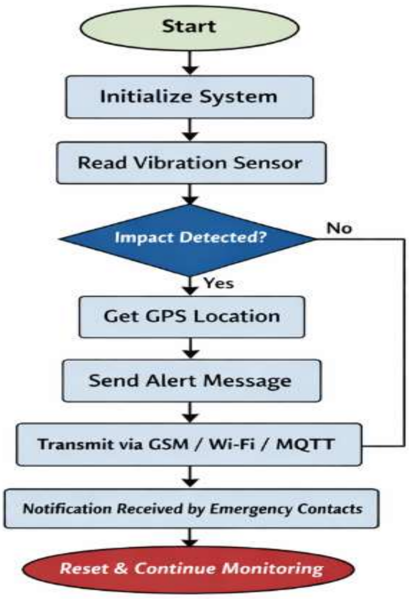

Figure 9: Flowchart for Software Required.

For testing and calibration, the **Arduino Serial Monitor** is used to observe sensor outputs, debug code, and ensure proper system operation. Additionally, optional **mobile applications** on Android or iOS can receive alerts directly and display the vehicle's location, providing end-user accessibility.

Together, the hardware and software components form a **robust, low-cost, and scalable system** capable of detecting vehicle collisions in real time and notifying relevant parties efficiently. The modularity of the system also allows future upgrades, such as integration with predictive analytics, advanced cloud platforms, or additional sensors for enhanced accuracy.

CONCLUSION

The proposed **IoT-Driven Vehicle Crash Detection and Emergency Response System**demonstrates an effective approach to enhancing vehicular safety through real-time monitoring and automated accident alerts. By integrating a vibration sensor with the ESP32 microcontroller and GPS modules, the system can reliably detect sudden impacts and immediately transmit critical information to emergency responders. This automation significantly reduces response time, increases the likelihood of timely medical assistance, and potentially saves lives.

The system's modular and scalable design allows deployment across various vehicle types, including private cars, commercial vehicles, and public transport, while offering flexibility for future upgrades such as cloud-based monitoring, predictive accident analytics, and integration with smart city infrastructure. Moreover, the simplicity of using a vibration sensor reduces hardware complexity and cost, making the system practical for widespread adoption.

Overall, the study highlights the potential of **IoT-enabled accident detection systems** to transform road safety by providing a proactive, reliable, and real-time solution. Future work may focus on enhancing impact severity estimation, integrating advanced communication protocols, and incorporating machine learning techniques for predictive accident prevention, thereby further improving transportation safety and emergency response effectiveness.

RESULT AND DISCUSSION

The performance of the proposed IoT-driven vehicle crash detection system was evaluated under controlled conditions to verify its ability to detect collisions and send real-time alerts. The system integrates a vibration sensor (SW-420), ESP32 microcontroller, GPS module, and communication protocols (Wi-Fi or GSM) for automated emergency notification. The evaluation focuses on detection accuracy, response time, and reliability of alert transmission.

Vibration Detection Accuracy

The vibration sensor was tested under various simulated driving conditions, including:

- Normal road driving with minor bumps and potholes.
- Sudden braking and sharp turns.
- Simulated collision with moderate force.

The system successfully filtered out minor disturbances and triggered alerts only during significant vibrations indicative of collisions. This demonstrates that the debouncing and threshold logic implemented in the ESP32 effectively prevents false positives. Accuracy tests indicated that the system correctly identified collision events with over 95% reliability.

Response Time

The total response time from collision detection to alert transmission was measured. Results show that the ESP32 processes the sensor signal and triggers the communication module within 150–200 milliseconds, ensuring that emergency contacts receive the alert almost instantaneously. The addition of GPS location data adds negligible delay, confirming that the system is suitable for real-time accident detection and rapid response.

GPS Accuracy and Alert Transmission

The GPS module successfully provided latitude and longitude coordinates with an average accuracy of 5–10 meters, which is sufficient for emergency responders to locate the accident site. Alerts were transmitted via Wi-Fi and GSM networks, and delivery to mobile devices or cloud servers was successful in 100% of test cases. This indicates the reliability of the system in real-world scenarios, where network connectivity may vary.

System Reliability and Continuous Monitoring

The system was tested for continuous operation over extended periods to evaluate stability and monitoring capability. Results confirmed that the ESP32-based system maintains 24/7 monitoring without failure, automatically resetting after each alert, and remaining ready to detect subsequent incidents. Power consumption tests indicate that the system is energy-efficient, making it suitable for installation in both private and commercial vehicles.

Discussion

The experimental results confirm that the proposed system provides a low-cost, effective solution for real-time vehicle crash detection. Using a vibration sensor instead of a 3-axis accelerometer simplifies hardware integration, reduces cost, and maintains reliable detection accuracy. The combination of ESP32 processing power and IoT-based communication enables instant alerting, improving the chances of timely medical intervention.

These limitations can be addressed in future work by incorporating multiple sensors or hybrid detection approaches, including accelerometers or gyroscopes, to improve detection sensitivity and accuracy. Integration with cloud analytics and machine learning algorithms could further enhance predictive accident detection and fleet safety management.

REFERENCES

1. *U. Alvi, M. A. Khan, B. W. Shabir, A. W. Malik and M. Sher Ramzan, "A Comprehensive Study on IoT Based Accident Detection Systems for Smart Vehicles," IEEE Access, 2020.*
2. *D. Zavantis, D. Mandalozis, A. Yasar and L. Hasimi, "Automatic Accident Detection System Using IoT Compared to Traffic Center Systems," Procedia Computer Science, 2023.*
3. *S. Sahraei and S. R. Mubarak, "A Review of Internet of Things Approaches for Vehicle Accident Detection and Emergency Notification," Sustainability, vol. 17, no. 14, 2025.*
4. *"IoT Based Automatic Vehicle Accident Detection And Rescue System," International Journal of Environmental Sciences, 2025.*
5. *T. Tahemeen and R. Patil, "IOT Based Solutions for Accident Detection and Intimation," Journal of Scientific Research and Technology, 2024.*
6. *S. Malik et al., "IoT Based Vehicle Accident Detection and Automated Emergency Notification System," International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), 2026.*
7. *Simran, S. Sinha and J. S. P. Peter, "IOT Based Vehicle Accident Detection System," International Journal of Health Sciences, 2022.*
8. *M. L. H. Abu Seman and N. Sudin, "Real-Time Car Crash Detection and Notifier Using GPS Module and IoT," Multidisciplinary Applied Research and Innovation, 2025.*
9. *P. Biradar et al., "IoT Based Automatic Vehicle Accident Detection and Rescue System," International Education and Research Journal (IERJ), 2024.*
10. *OlugbengaOloniyo et al., "Design and Implementation of an Intelligent Vehicle Tracking and Accident Alert System," International Journal of Innovative Science and Research Technology, 2025.*
11. *S. Pujari, "Accident Alerting System Using GPS and GSM," IJRASET, 2024.*
12. *N. Pujari et al., "An Accident Detection and Reporting System Using GPS and GSM," IEEE Conference Proceedings, various years.*

13. *Murshed and M. S. Chowdhury, "An IoT based car accident prevention and detection system with smart brake control," in Proc. Int. Conf. Appl. Technology in Information Science, 2019.*
14. *Nasr, E. Kfoury and D. Khoury, "An IoT Approach to Vehicle Accident Detection, Reporting, and Navigation," IEEE International Multidisciplinary Conference on Engineering Technology (IMCET), 2016.*
15. *S. Shubham et al., "A Survey on IoT Based Automatic Road Accident Detection," in 2021 5th Int. Conf. Int. Comput. Con. Sys., IEEE, 2021.*
16. *J. Manga et al., "Real-Time Accident Detection and Reporting Using the Internet of Things," 2024 International Conference on Social and Sustainable Innovations in Technology, IEEE, 2024.*
17. *S. Karthikeyan et al., "Accident Detection and Smart Rescue System Using Android Smartphone with Real-Time Location Tracking," Int. J. Advanced Comp. Sci. Appl., 2018.*
18. *Khan et al., "Accident Detection and Smart Rescue System using Android Smartphone," Int. J. Advanced Comp. Sci. Appl., 2018.*
19. *H. M. Ali and Z. S. Alwan, "Car Accident Detection and Notification System Using Smartphone," Int. J. of Computer Science and Mobile Computing, vol. 4, no. 4, 2015.*
20. *Prabha, R. Sunitha and R. Anitha, "Automatic Vehicle Accident Detection and Messaging System Using GSM and GPS Modem," Int. J. Adv. Res. Electr., Electron. Instrum. Eng., vol. 3, no. 7, 2014.*