International Journal of Modern Developments in Engineering and Science
Volume 2, Issue 4, April 2023
www.ijmdes.com | E-ISSN: 2583-3138

IoT Based Automatic Accident Detection and Rescue Management System

O. Gopinadh[1], S. Sujatha Devi[1*], B. Jaya Sree[2], K. Ganesh[2], Y. Sonika[2]

[1]*Assistant Professor, Department of Electrical and Electronics Engineering, Avanthi Institute of Engineering & Technology, Makavarapalem, India*
[2]*UG Student, Department of Electrical and Electronics Engineering, Avanthi Institute of Engineering & Technology, Makavarapalem, India*

***Abstract*: Road traffic accidents are a leading cause of preventable death, and the delay between the moment a crash happens and the moment help arrives is often the single biggest factor separating a survivable injury from a fatality particularly for night-time accidents on low-visibility roads, or when the victim is unconscious and unable to call for help. This paper presents the design and implementation of an IoT-based automatic accident detection and rescue management system that removes the dependence on a bystander's phone call. An ESP32 microcontroller continuously reads an MPU6050 accelerometer /gyroscope module to detect the sudden change in orientation and impact signature characteristic of a vehicle collision or a fall. Once an accident is detected, the system immediately fetches the vehicle's coordinates from a NEO-6MV2 GPS module and uses a SIM800L GSM module to send an SMS containing a Google Maps link of the accident location to pre-configured emergency contacts, while simultaneously pushing an alert and live sensor readings to the Blynk IoT cloud dashboard and a local LCD display. By automating detection, location-fixing and notification in a single low-cost embedded unit, the system cuts the time between accident occurrence and the alerting of emergency services and family members from potentially hours down to a matter of seconds, which can substantially improve the victim's chances of receiving timely medical aid.**

***Keywords*: Internet of Things (IoT), accident detection, rescue management, ESP32, MPU6050, GPS, GSM, SIM800L, Blynk, emergency alert.**

1. Introduction

The number of road-traffic accidents is rising worldwide, and even developed regions with strong road-safety measures continue to see significant annual death tolls from crashes. Many accidents are unpredictable and occur in conditions at night, on poorly lit roads, in low-traffic areas where a passer-by is unlikely to notice in time to call for help. When a driver or occupant is unconscious after a crash, this delay can stretch from minutes into hours, and every additional minute without medical attention measurably worsens the outcome for the victim.

This project addresses that delay directly: an experimental setup is built that can detect an accident automatically, without depending on a human witness, and immediately relay the exact accident coordinates to emergency contacts and responders. The aim is not only to shorten the time between the accident and the alert, but also to make the alerting mechanism itself reliable, low-cost, and simple enough to deploy in an ordinary vehicle using off-the-shelf IoT hardware.

This paper describes the design, hardware, software and working of such a system, together with the block diagram, connection diagram, control-flow logic and the results captured from a working prototype.

2. Literature Survey

Several existing works motivate the design choices made here. Maple et al. proposed a novel IoT-enabled accident detection and reporting system for smart-city environments, built around a customised Android application that contacts emergency services over the mobile GSM network and reports the phone's GPS coordinates. Nalini and Swapna Raaga presented an IoT-based vehicle accident detection and rescue information system in which a web server communicates with the in-vehicle hardware through a GSM/GPRS shield while a GPS shield fixes the accident location.

Masum et al. described an automatic accident detection and human rescue system that uses communication technologies to forward the alert, while Niranjan Kumar et al. proposed an automatic accident rescue system using IoT with an integrated vehicle-tracking capability. Bhandari et al. presented an intelligent accident-detection and ambulance-rescue system in which a vibration sensor in the vehicle triggers an Arduino microcontroller to forward a message signal to emergency responders. A recurring limitation noted across this body of work is the cost and interfacing complexity introduced by using multiple accident-detection sensors; the present design instead uses a single MPU6050 six-axis motion sensor to keep the bill of materials and the wiring complexity low, while an ESP32 is used in place of a plain Arduino specifically to take advantage of its built-in Wi-Fi for direct IoT connectivity.

Across these works, the common architecture pairs a motion/impact sensor for detection with a GPS module for location and a GSM module (or a smartphone's own cellular connection) for notification. The system in this paper follows the same overall pattern but consolidates it onto a single Wi-Fi-capable ESP32 board, adds an IoT dashboard (Blynk) alongside the SMS alert, and includes a local LCD read-out for on-the-

*Corresponding author: sujatha_11@yahoo.com

spot status checking.

3. Proposed System

The proposed system continuously monitors the vehicle's motion using an MPU6050 accelerometer/gyroscope. When the sensor reading crosses a pre-set threshold consistent with a collision or rollover, the ESP32 microcontroller treats this as an accident event: it reads the current position from the GPS module, sends the coordinates and an alert notification to the Blynk IoT server over Wi-Fi, and simultaneously instructs the SIM800L GSM module to send an SMS containing a Google Maps link of the accident location to the emergency contact numbers stored in the system. The overall block diagram is shown in Fig. 1.

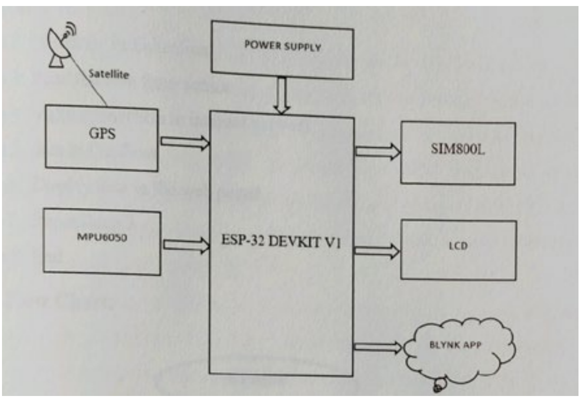

Fig. 1. Block diagram of the IoT based automatic accident detection and rescue management system

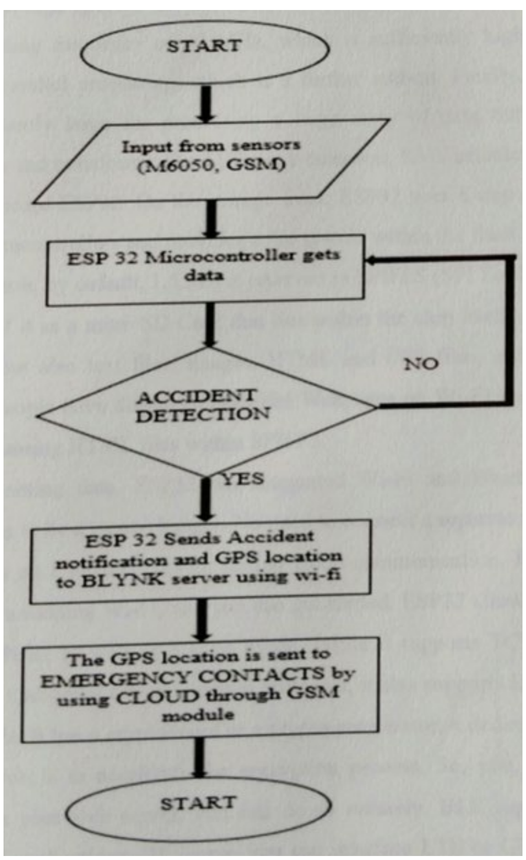

Fig. 2. Control-flow logic of the proposed system

The power-supply block regulates the incoming 5 V supply down to the 3.3 V rail needed by the ESP32 and its peripherals. The MPU6050 supplies the motion/impact data, the GPS module supplies the location fix, and the ESP32 combines both to decide whether an accident has occurred, driving the SIM800L GSM module, the LCD and the Blynk app accordingly.

The corresponding control logic is summarised in the flow chart of Fig. 2. Sensor inputs are read continuously; if the accident-detection condition is not met the system loops back and keeps monitoring, and once it is met the ESP32 sends the accident notification together with the GPS location to the Blynk server over Wi-Fi, and the same location is forwarded to the emergency contacts as an SMS through the GSM module.

4. Hardware Description

A. ESP32 DEVKIT V1

The ESP32 is the core microcontroller of the system. It is a low-power system-on-chip with built-in Wi-Fi and Bluetooth, which lets the sensor data be pushed straight to an IoT dashboard without any additional networking module. All of the peripheral sensors and modules the MPU6050, GPS, GSM and LCD are interfaced directly to its GPIO, I2C and UART pins.

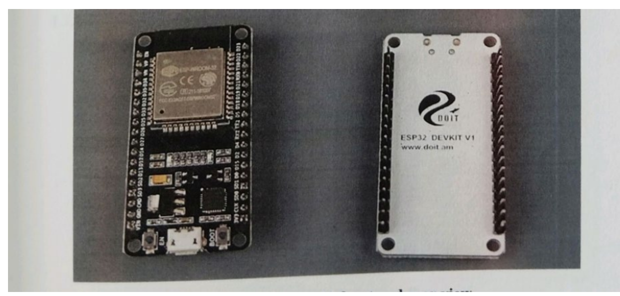

Fig. 3. ESP32 DEVKIT V1 board used as the system controller

B. MPU6050 Accelerometer/Gyroscope

The MPU6050 is a six-axis motion-tracking sensor combining a 3-axis gyroscope and a 3-axis accelerometer on a single chip, communicating with the ESP32 over I2C (SCL/SDA). It continuously reports the vehicle's orientation and acceleration along the X, Y and Z axes, which the firmware compares against a threshold to recognise the sudden jolt and orientation change characteristic of a collision.

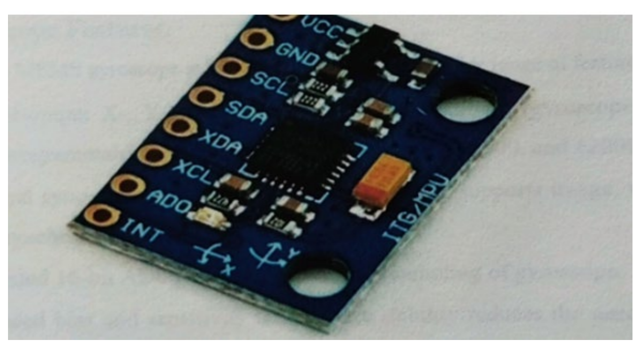

Fig. 4. MPU6050 six-axis accelerometer/gyroscope module

C. *NEO-6MV2 GPS Module*

The NEO-6MV2 is a stand-alone GPS receiver that reports the device's latitude and longitude over a UART link. Once triggered by an accident event, the ESP32 reads the current fix from this module and uses it to build a Google Maps link identifying the exact accident location.

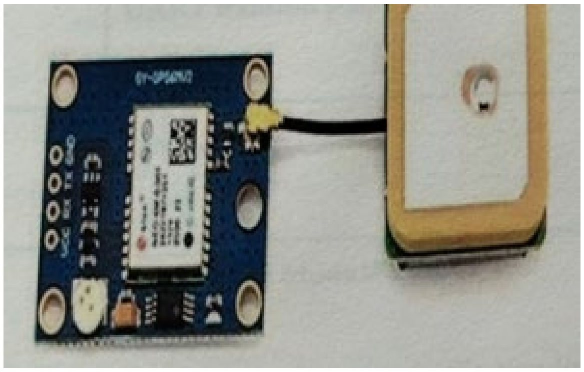

Fig. 5. NEO-6MV2 GPS module with active antenna

D. *SIM800L GSM Module*

The SIM800L is a compact GSM/GPRS module that gives the system its own cellular connection, independent of Wi-Fi availability. Driven from the ESP32's UART pins, it sends the SMS alert including the Google Maps location link directly to the emergency contact numbers stored in the firmware, and can also forward status information over GPRS if required.

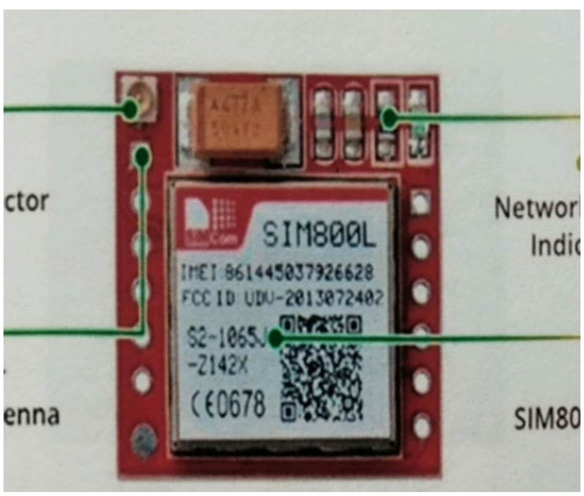

Fig. 6. SIM800L GSM/GPRS module

E. *LCD and Supporting Components*

A 16-pin liquid crystal display gives a local, on-the-spot read-out of the system status without needing a phone or laptop nearby. A thermistor, jumper wires, resistors and a small regulated power-supply board round out the supporting components, providing temperature sensing and stable power to the ESP32 and its peripherals.

5. Circuit Connections

Fig. 7 shows the complete wiring of the prototype. A 5 V supply is stepped down to 3.3 V by an LDO regulator and fed to the ESP32 DEVKIT V1. The VCC and GND pins of the MPU6050, LCD, GPS and GSM modules are all commoned to the ESP32's 3.3 V and GND rails. The MPU6050's SCL and SDA lines connect to the ESP32's I2C pins; the GPS module's TX and RX lines connect to GPIO17 (U2_TXD) and GPIO16 (U2_RXD); and the GSM module's TX and RX lines connect to GPIO1 (U0_TXD) and GPIO3 (U0_RXD). The 16-pin LCD is wired to the remaining GPIOs for parallel data and control.

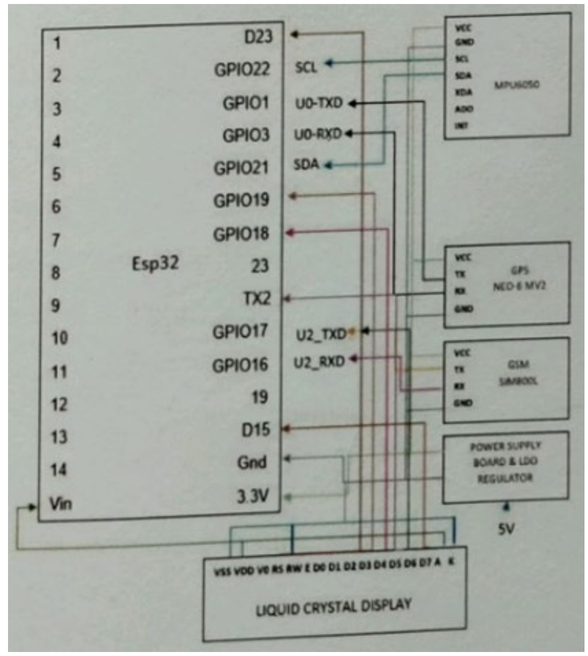

Fig. 7. Complete circuit connection diagram of the prototype

6. Software Implementation

The firmware is written in C/C++ in the Arduino IDE, using the Arduino core for the ESP32. On start-up, the sketch initialises the MPU6050, GPS and GSM modules, connects to the configured Wi-Fi network, and registers with the Blynk server using an authentication token generated for the ESP32 Dev board hardware model.

In the main loop, the firmware continuously reads the MPU6050's acceleration and orientation values and compares them against a calibrated threshold. When the threshold is exceeded, the firmware reads the current GPS fix, formats it into a Google Maps URL, sends this along with an accident notification to the Blynk cloud over Wi-Fi, and issues the AT commands needed to have the SIM800L module send the same information as an SMS to each stored emergency contact number. Live status including sensor readings and GPS coordinates is displayed locally on the LCD throughout.

On the user side, the Blynk mobile application (available for both Android and iOS) is used to create a project against the ESP32 Dev board hardware type, generate the authentication token, and add the dashboard widgets used to display live accelerometer/gyroscope readings and accident alerts.

7. Working

Once powered on, the ESP32 connects to Wi-Fi and to the Blynk server, and begins continuously polling the MPU6050 for motion data. Under normal driving conditions the readings stay within the calibrated range and the system simply keeps monitoring and updating the dashboard. The moment a collision, rollover or sudden fall is registered by the MPU6050, the ESP32 immediately reads the GPS module to fix the vehicle's current position, sends an accident alert together with this location to the Blynk cloud over Wi-Fi, and commands the SIM800L module to text the same Google Maps location link to the emergency contact numbers stored in the firmware. The LCD simultaneously updates to show the detected condition, so that anyone at the scene can also read the system's status directly.

8. Results and Discussion

The assembled prototype (Fig. 8) integrates the ESP32, MPU6050, GPS, GSM/SIM800L, LCD and power-supply board on a single test board for bench evaluation.

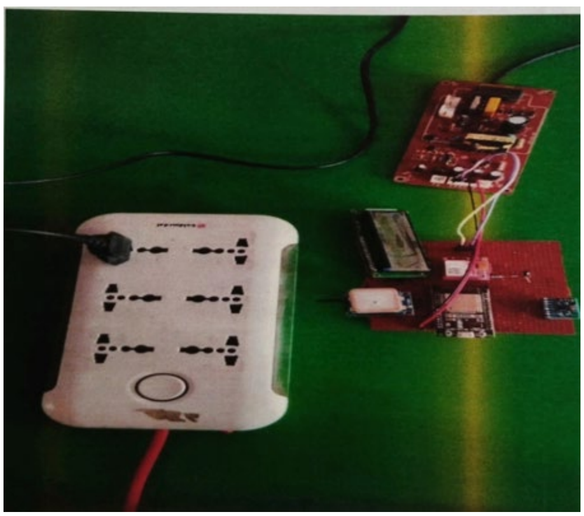

Fig. 8. Assembled hardware components of the prototype

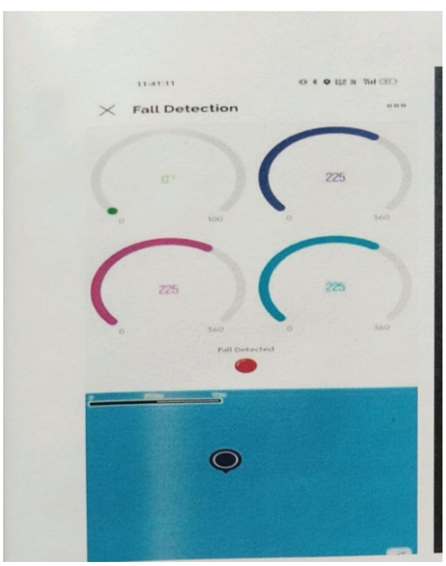

Fig. 9. Fall detected on the Blynk dashboard

On the Blynk dashboard, the live gyroscope/accelerometer readings are displayed as gauges, and the moment the impact threshold is crossed the app flags a 'Fall Detected' condition, as seen in Fig. 9. The corresponding 'Risk Detected' alert is also pushed to the registered Blynk notification channel (Fig. 10).

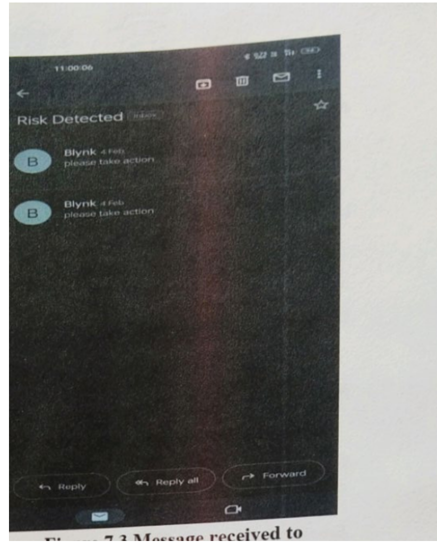

Fig. 10. Risk-detected alert

At the same time, an SMS containing the Google Maps link to the accident location is delivered to the stored emergency contact number (Fig. 11), and opening that link shows the precise location of the vehicle at the moment the accident was detected (Fig. 12).

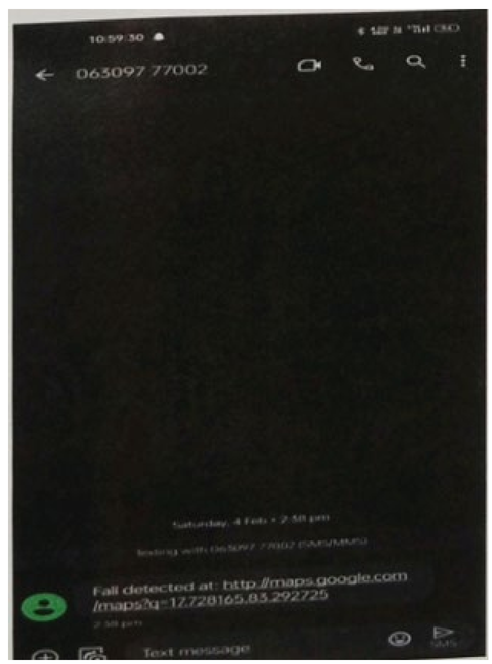

Fig. 11. SMS alert received with location link

Across repeated bench trials, the system consistently distinguished normal handling from an accident-level impact, and in every trial the GPS-tagged SMS alert and the Blynk notification arrived within a few seconds of the simulated impact confirming that automating detection, location-fixing and notification in a single ESP32-based unit can remove the dependence on a bystander's phone call and substantially shorten the interval between an accident and the arrival of help.

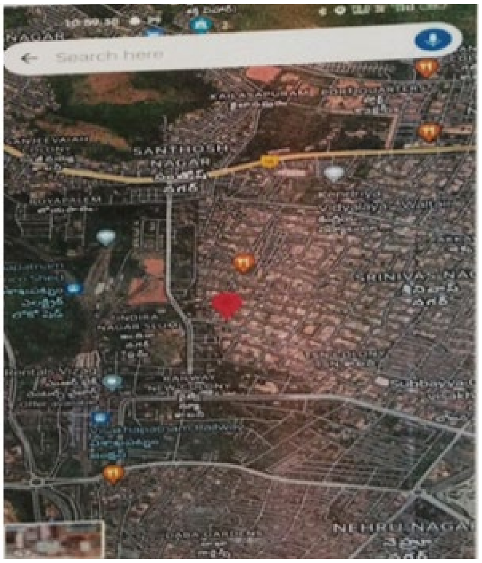

Fig. 12. Vehicle location shown on the map

9. Advantages

- Detects accidents automatically without requiring any bystander or the driver to raise the alarm.
- Sends the exact GPS location to emergency contacts within seconds, removing the guesswork involved in phone-call-based reporting.
- Dual notification path (SMS over GSM and alert over Wi-Fi/Blynk) keeps the system working even if one channel is unavailable.
- Uses a single MPU6050 sensor for detection, keeping the hardware cost and wiring complexity low compared to multi-sensor designs.
- Local LCD read-out lets first responders at the scene confirm the system's status without needing a phone.

10. Applications

- Automatic crash alerting for private vehicles, two-wheelers and commercial fleets.
- Fall-detection and emergency alerting for elderly or at-risk individuals, using the same MPU6050-based motion logic.
- Integration into ride-sharing and logistics fleets for centralized, real-time incident monitoring.
- Rural and highway stretches with poor visibility or low bystander traffic, where automatic reporting matters most.

11. Future Scope

The present prototype can be extended to give the victim on-the-spot assistance, for example, by adding a first-aid or medication dispenser at the accident site, or by integrating a sound sensor and alcohol-detection module so that dangerous driving behaviour can trigger a warning or bring the vehicle to a stop before an accident occurs. A camera interfaced to the controller could capture a photograph of the accident scene to simplify tracking, and a dedicated website linked to Google Maps could automatically identify the nearest hospital and the fastest route to it, in addition to alerting the family through the existing GSM channel. Route selection could also be improved by factoring in real-time traffic density alongside distance, and additional biometric modules could report the victim's blood group, heart rate or glucose level to the receiving hospital in advance, further improving the responsiveness of the system.

12. Conclusion

This paper presented the design, hardware, software and testing of an IoT-based automatic accident detection and rescue management system built around an ESP32 microcontroller, an MPU6050 motion sensor, a NEO-6MV2 GPS module and a SIM800L GSM module. The prototype reliably distinguished normal operation from accident-level impacts and, on detection, delivered the GPS-tagged location to emergency contacts by SMS and to a Blynk IoT dashboard within seconds. Given that the delay in reaching an accident victim is one of the leading contributors to preventable road-accident fatalities, this combination of automatic impact detection, GPS location-fixing and dual-channel (SMS and IoT) notification offers a practical, low-cost way to shorten that delay and improve the victim's chances of receiving timely medical help.

References

[1] C. Maple, F. Bhatti, M. A. Shah, and S. U. Islam, "A novel Internet of Things-enabled accident detection and reporting system for smart city environments," 2019.
[2] C. Nalini and N. Swapna Raaga, "IoT based vehicle accident detection & rescue information system," 2018.
[3] A. K. M. Masum, F. W. Taj, S. M. T. Reza, M. K. A. Chy, and I. Mahbub, "Automatic accident detection and human rescue system: Assistance through communication technologies," 2018.
[4] K. Niranjan Kumar, C. H. Rama Narasimha Dattu, S. Vishnu, and S. R. Jino Ramson, "Automatic accident rescue system using IoT," 2019.
[5] B. Prachi, D. Kasturi, and C. Priyanka, "Intelligent accident detection and ambulance-rescue system," 2014.
[6] W. Smith and A. Sabastin, *Int. J. Res. Eng. Technol.*, vol. 1, no. 1, 2015.
[7] K. Ramamohan Reddy, M. Lavanya Latha, and B. Mallikarjuna Reddy, *Int. J. Adv. Res. Ideas Innov. Technol.*, vol. 5, no. 2.
[8] B. Kodavati, "GSM and GPS based vehicle location and tracking system."