International Journal of Wireless and Ad Hoc Communication

♦ ISSN: 2692-4056 Vol: 09, No. 01, 2025 ♦ PP. 25–30

Article DOI: https://doi.org/10.54216/IJWAC.090104

FULL LENGTH ARTICLE

Smart Accident Detection using IoT Technology

Sindhuja M.[1,*] **Vijay Murugan S.**[2] **Elarmathi S.**[3]

[1] Assistant Professor, School of Electronics, Vellore Institute of Technology, Chennai, Tamil Nadu, India
[2] Assistant Professor, Department of ECE, Paavai Engineering College, Namakkal, Tamil Nadu, India
[3] Assistant Professor, Department of ECE, Knowledge Institute of Technology, Salem, Tamil Nadu, India

Emails: sindhuja.m@vit.ac.in · vijaymuruganeee@gmail.com · elarmathivijay@gmail.com

Received: November 04, 2024 Revised: December 27, 2024 Accepted: January 18, 2025 ⋆ Corresponding author

ABSTRACT

Road accidents and emergency-service delay are significant issues. Efficient accident handling through immediate detection and timely aid is therefore crucial. Accident detection and emergency systems based on Internet of Things (IoT) technology, with minimum delay, are gaining significant attention in industry and academic literature. In this work, an effective accident-detection method is proposed by employing five sensors not only to detect an accident but also to report the accident type, such as collision, no accident, rollover, or fall-off. In addition, accident status is communicated to the IBM Watson Cloud platform. Incoming data received in the Node-RED platform are integrated with Google Maps to show the location and other accident information that can be accessed by hospitals through a website, while alert messages are sent to the victim's acquaintances. Two machine-learning models, K-Nearest Neighbor (KNN) and Naive Bayes (NB), are compared to identify the best accident-detection model. The KNN model is observed to be the most effective ML model for identifying accident status and enhancing the system by providing patient details, a kill switch, and repeated alert messages until acknowledgement is received.

Keywords: Internet of Things ▪ Accident detection ▪ Machine learning ▪ Sensors ▪ Collision ▪ Emergency

1. INTRODUCTION

Nowadays, the number of accidents occurring worldwide rises with increases in population and vehicle count. A recent World Health Organization estimate indicates that 50 million people are injured and 1.35 million people die annually. Delays in receiving emergency medical aid result in human casualties in traffic collisions. Reducing the response time of emergency medical care may lessen the likelihood of death.

IoT is one of the most promising information and communication technologies used to reduce accident response time. It provides an interconnected network of physical components and devices to track and control automobiles. Significant research has been carried out to detect accidents and reduce post-accident response time [1].

2. LITERATURE REVIEW

Kumar et al. [2] developed an IoT-based automobile accident detection and categorization (ADC) system that identifies and reports accident type through sensor fusion. Three ADC models were assessed and contrasted. With an average F1-score of 0.95, the NB model was found to perform better than the others, with its highest accuracy observed for collision detection.

Sherif et al. [3] reported a real-time traffic accident warning system employing wireless sensor networks and Radio-Frequency Identification technologies. Shaik et al. [4] presented automatic notification and response using IoT, where accelerometer and GPS signals are transferred to the cloud and alarm messages are delivered to subscribers. Anil et al. [5] used flex sensors and an accelerometer to identify

© 2025 The Author(s). Published by ASPG. This article is licensed under CC BY 4.0.

accidents and send nearby hospital-location information.

Rakhonde et al. [6] implemented accident detection together with accident prevention and pollution monitoring. Verma et al. [7] compared frame size and time consumption in accident detection and used Canny-based edge detection for faster accident-pattern detection. Karmokar et al. [8] proposed a smart accident detection and rescue mechanism that provides the exact accident location and helps ambulances take the shortest path to the accident spot. VANET-based approaches communicate alert messages through radio-frequency modules when accidents occur [9, 10].

Smartphone-based systems use Android applications to identify auto crashes by measuring speed and tilt angle through accelerometer and GPS units [11]. Other approaches collect data from sensors and event logs, then employ computational models such as nearest-neighbor classifiers, neural networks, and regression trees [12].

The contributions of this work are as follows:

- Receive inputs from multiple sensors.
- Feed information into the selected model to determine whether an accident has happened.
- Immediately share the location with the closest hospital for ambulance dispatch.
- Report both accident occurrence and accident type, including crash, rollover, and fall-off.
- Alert loved ones subscribed to the vehicle.
- Enable a kill switch if the accident is not severe.

3. PROPOSED METHODOLOGY

Road Transport Minister Nitin Gadkari has stated that India faces a bigger threat from road accidents than the COVID-19 pandemic. Although India accounts for around 1% of the world's vehicles, it contributes about 11% of global road-accident deaths according to a World Bank report. Of the 450,000 accidents that happen annually, 33.3% of people die. A person dies every four minutes from a road accident, according to the same report.

Trauma specialists identify a golden period: the first 60 minutes after an accident. The care administered to the patient and the subsequent transport to the nearest hospital can make a decisive difference. Therefore, the proposed method aims to minimize this gap as much as possible.

3.1 Detection System

The accident-detection system begins with three primary sensors selected according to system requirements. The MPU6050 provides accelerometer and gyroscope values simultaneously, reducing size and cost. Vibration sensors are placed on four sides of the vehicle; they cover a wider region than traditional crash switches and reduce the chance of missing impacts. A flame sensor detects fire, since fires may occur after accidents.

Sensor values are collected by the Arduino Uno microcontroller and serially transmitted to the Raspberry Pi 3. The Raspberry Pi also reads the current GPS position and remains

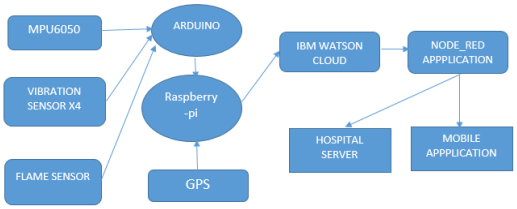


Figure 1. Accident detection system.

ready to send the location. Machine-learning models are executed on the Raspberry Pi to analyze accuracy and select the best model for classifying accident types.

If an accident occurs, the Raspberry Pi detects it immediately and sends details such as accident type and location to the IBM Watson Cloud. From there, information is transferred to the Node-RED application, which alerts hospital servers and family members through an application created using MIT App Inventor.

3.2 Component Descriptions

Figure 2 shows the interfacing of Arduino UNO with the vibration sensor, MPU6050 sensor, and flame sensor.

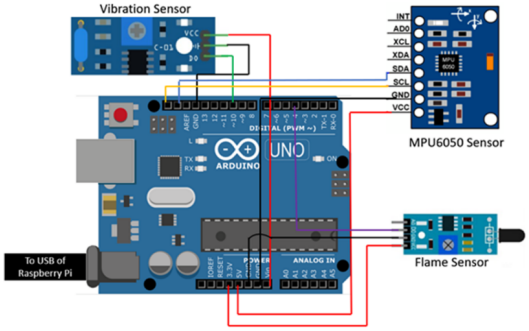


Figure 2. Arduino UNO interface with sensors.

3.2.1 MPU6050

The MPU6050 sensor measures accelerometer and gyroscope properties across three axes. Only Vcc, Gnd, serial clock, and serial data pins are required.

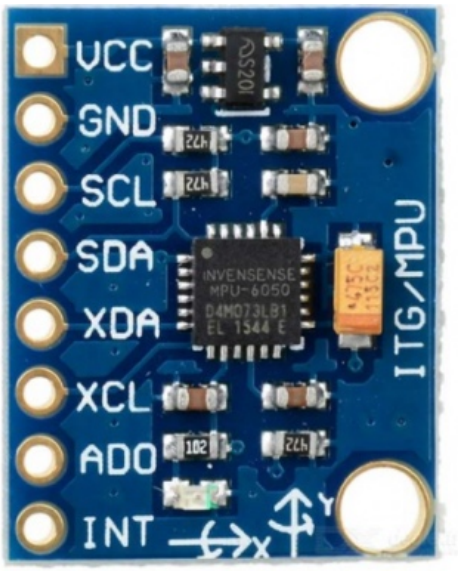


Figure 3. MPU6050 sensor.

3.2.2 Accelerometer

If a car becomes unstable due to an accident, its orientation changes. This can be detected by the accelerometer because the vehicle no longer remains parallel to the gravitational axes [13]. During an accident, the car decelerates and impact is distributed along three axes. Absolute linear acceleration (ALA) is calculated as

$$\text{ALA} = \sqrt{A_x^2 + A_y^2 + A_z^2}. \quad (1)$$

3.2.3 Gyroscope

The gyroscope provides pitch, roll, and yaw values. In this work, roll and pitch are emphasized because values exceeding 90 degrees may indicate rollover. Combining accelerometer and gyroscope readings provides required information about vehicle orientation.

3.2.4 Vibration Sensor

The vibration sensor identifies impact or collision experienced by the vehicle [11]. Its sensitivity is adjusted so that it detects changes only when the vehicle is hit hard.

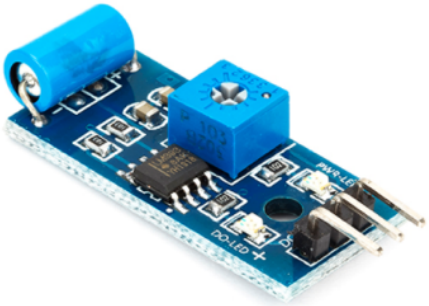


Figure 4. Vibration sensor.

3.2.5 Flame Sensor

When a crash occurs, fire may be anticipated due to gas presence and sparking. In such cases, paramedics and firefighters can be alerted if needed [12].



Figure 5. Flame sensor.

3.2.6 Pressure Sensor

A pressure sensor is included to calculate altitude change if a vehicle falls from a cliff or bridge. Atmospheric pressure assists in calculating change in height.

3.2.7 Arduino UNO

The Arduino Uno collects required data and communicates these details to the Raspberry Pi, which performs the heavy processing for machine-learning models [14]. The Arduino has sufficient Vcc pins and digital I/O pins for the sensors.

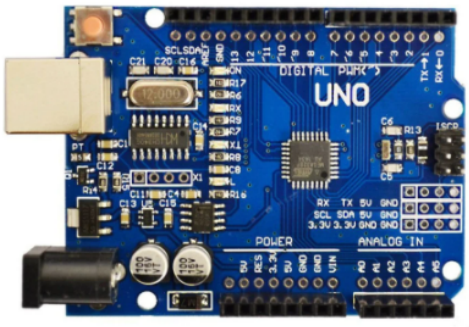


Figure 6. Arduino Uno.

3.2.8 Raspberry Pi

The Raspberry Pi 3B+ processes sensor data and GPS/GSM data. It can handle software applications such as machine learning, and it communicates accident location, driver details, and related information to IBM Watson Cloud [15].



Figure 7. Raspberry Pi.

3.2.9 GPS

The GPS sensor provides inputs in the form of NMEA messages. These messages include longitude, latitude, speed, altitude, and other values [16].

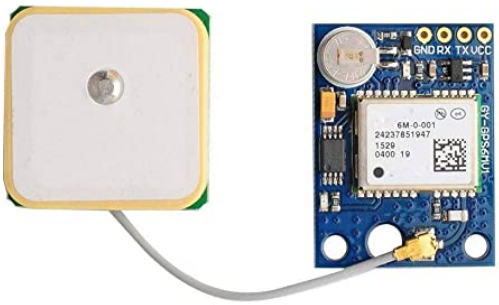


Figure 8. GPS module.

4. IMPLEMENTATION OF ACCIDENT DETECTION DEPENDS ON IOT

4.1 Collecting the Dataset

Dataset gathering uses properties arriving from various Arduino ports. Accelerometer values from the MPU6050 are normalized to ensure that the model remains relevant if data change with respect to vehicle location. Gyroscope values provide pitch, roll, and yaw, which help determine rollover. The vibration sensor serves as the impact sensor and indicates collision. Three outputs are used: 0 for no accident, 1 for collision, and 2 for roll-off. Arduino sends these values to the serial monitor, from which data are streamed into Excel

using the Data Streamer tool and then randomized for model training. Sample collected data are shown in Tables 1 and 2.

Table 1. Dataset collection with outputs: no accident and collision.

AX	AY	AZ	ALA	Temperature	GX	GY	GZ	Vibration	Output
-1472	564	13812	13901.66263	38.74	-468	-60	-57	0	0
-924	328	18048	18074.6138	38.69	-409	-93	24	0	0
-1032	-1016	6200	6366.88935	38.60	-596	174	-117	0	0
23220	25184	32767	47403.36006	38.60	-3624	18632	-357	896	1
3124	-29368	32767	44112.52757	38.69	11193	790	-523	895	1
-3152	-12528	1204	12974.41729	38.68	-4687	-2405	965	0	0
-3956	2256	6472	7913.675252	38.69	542	627	3563	774	1
-3176	1976	25960	26228.09852	38.65	5543	2381	4295	697	1
-7476	-556	21404	22678.86523	38.65	-4269	-59	3943	477	1

Table 2. Dataset collection with output: rollover.

AX	AY	AZ	ALA	Temperature	GX	GY	GZ	Vibration	Output
44	-1894	-4876	19565.37799	38.93	-7082	-1081	1077	897	2
-2360	-1704	-1040	17234.0593	38.84	-8410	-1323	-2920	0	2
-1744	-1739	1828	17578.50779	38.84	-4110	-1497	7092	897	2
-8344	-1763	-1429	24184.40357	38.88	-3917	-3940	409	897	2
-1424	-3276	-1166	34809.84344	38.84	-9774	12	-6113	897	2
-2690	32767	-3276	53583.98389	38.93	32767	1828	11387	885	2
-8	-2029	-1016	22693.41155	38.88	-32768	17559	-2243	895	2

4.2 Processing Data Using Machine-Learning Models

During dataset creation, accelerometer values are too random for direct interpretation because they contain three axes. A normalization tool is therefore needed to use the magnitude of acceleration rather than directions. Applied linear acceleration makes it easier for the machine to detect an accident because definite deceleration occurs regardless of direction. Since the dataset contains output labels, supervised-learning algorithms are used.

4.3 Sending Information through Cloud to Hospitals Using Node-RED

This part has two tasks: collecting coordinates from GPS and communicating this information to the cloud.

4.3.1 Connecting GPS to Obtain Longitude and Latitude

After obtaining a GPS fix using U-center software, the GPS connects to seven satellites and provides accurate readings. The GPS is connected to one of the GPIO ports in the Raspberry Pi.

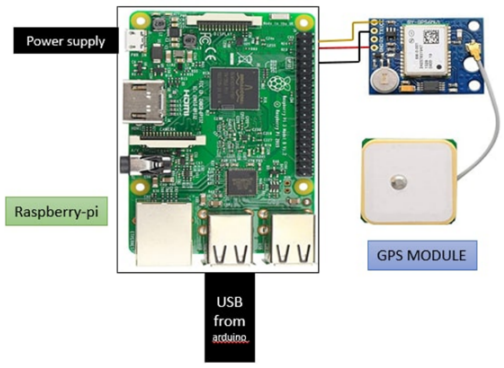


Figure 9. GPS connection with GPIO port.

The information is received serially by the Raspberry Pi, but formatting is required to extract latitude and longitude. Only the `$GPGGA` sentence is required from the NMEA sentences. The code extracts the needed values and sends them to the cloud with other information such as accident status and fire presence.

4.3.2 Connecting to IBM Watson

The IBM Watson Cloud provides tools for the system workflow. A Cloudant database is created from the Watson catalog to store information in JSON format. After assigning an instance name and selecting IAM authentication, the database is deployed. The IoT platform is then launched from the resource list, a device is created, and sensor values are received as recent events.

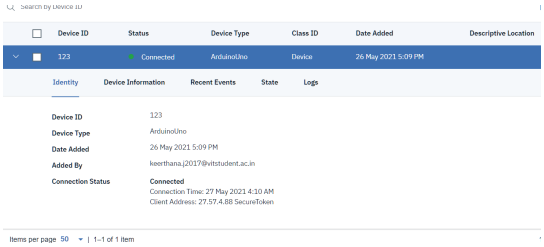


Figure 10. Connection made between Raspberry Pi and cloud.

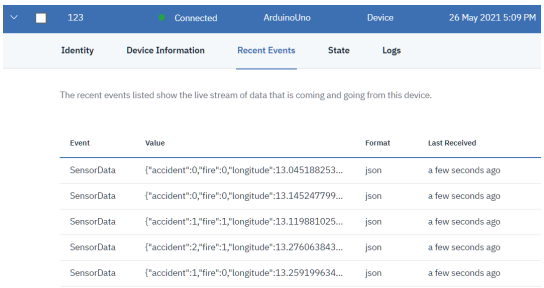


Figure 11. Values being received in the cloud.

After verifying that values are conveyed from the device, a standard application is created and API details are recorded. A website is then built using the Node-RED application in the IBM Watson Platform. Figure 12 shows the flow used for the website.

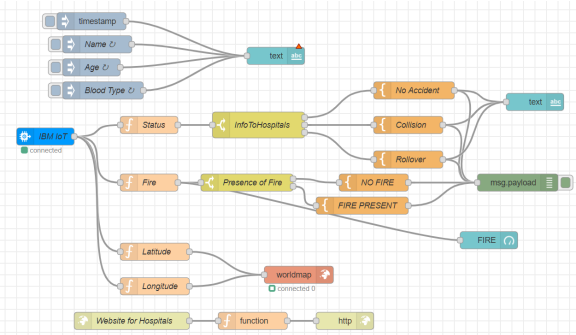


Figure 12. Node-RED flow for using data and sending it to the hospital server.

Values received in JSON format are extracted into individual properties and compared with thresholds. The corresponding message is sent to the hospital and loved ones. The details and position on the map are displayed using UI and world-map features.

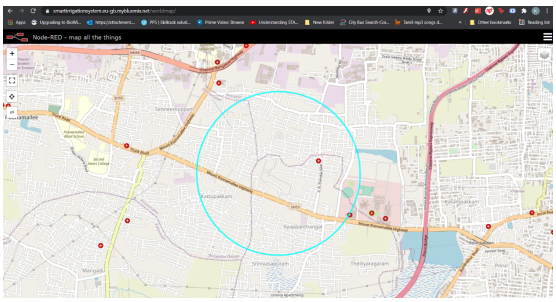


Figure 13. Node-RED map image.

Fast2SMS is used as an SMS service provider for emergency messages, depending on whether an accident has occurred. It

is activated using an authorization key and token.

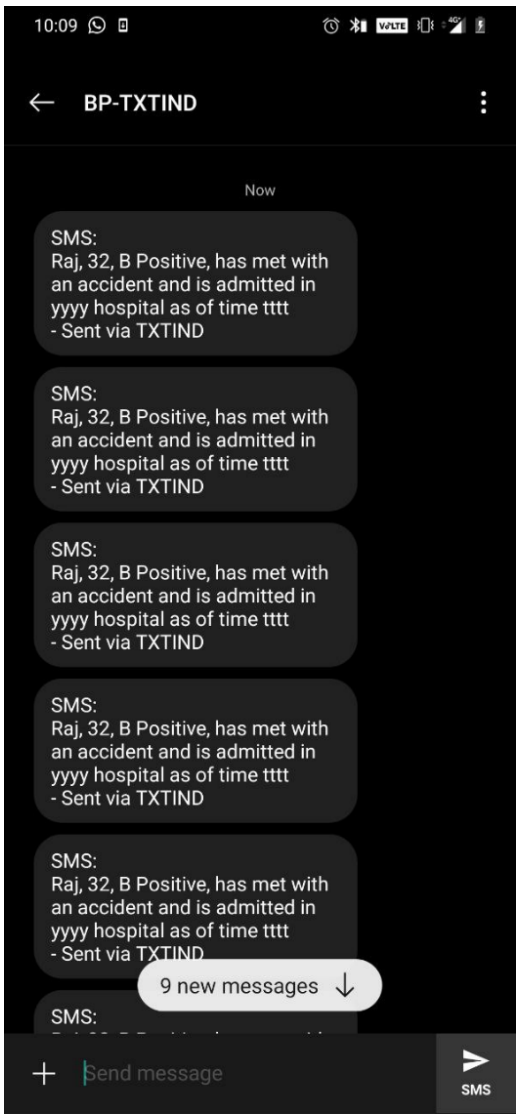


Figure 14. Emergency message to mobile phone.

5. RESULTS AND DISCUSSION

Figure 15 and Table 3 show Arduino output, transfer of sensor values to the Raspberry Pi, and the dataset collected using Data Streamer in Excel.

Table 3. Dataset collected using Data Streamer in Excel.

AX	AY	AZ	ALA	Temperature	GX	GY	GZ	Vibration	Output
1472	564	13812	13901.66263	38.74	-468	-60	-57	0	0
-924	328	18048	18074.6138	38.69	-409	-93	24	0	0
-1032	-1016	6200	6366.88935	38.60	-596	174	-1179	0	0
23220	25184	32767	47403.36006	38.60	-3624	18632	-3571	896	1
3124	-2936	32767	44112.52757	38.69	11193	790	-5235	895	1
-3152	-1252	1204	12974.41729	38.69	-4687	-2405	965	0	0
-3956	2256	6472	7913.675252	38.69	542	627	3563	774	1
-3176	1976	525960	26228.09852	38.65	5543	2381	4295	697	1
-7476	-556	21404	22678.86523	38.65	-4269	-59	3943	477	1
-3812	6412	-512	7477.113882	38.74	-1969	-25	3978	0	0
-828	2088	12860	13054.68989	38.69	-2978	1524	1266	0	0
-3936	4360	9824	11446.07671	38.74	-3340	809	2293	0	0

When the car is idle or moving without an accident, the system output is shown in Figure 16.

When the car goes through a collision while moving, the output is shown in Figure 17.

When the car has rolled over due to an accident while moving, the output is shown in Figure 18.

6. CONCLUSION AND FUTURE SCOPE

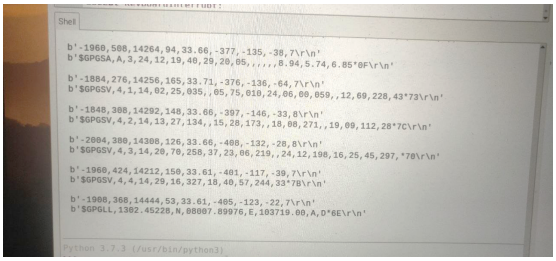
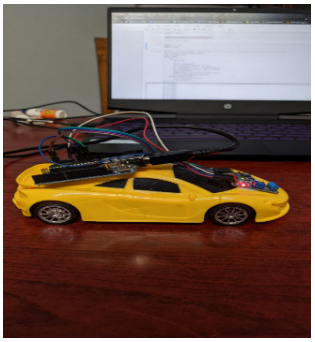


Figure 15. Output of Arduino transferring sensor values to the Raspberry Pi.



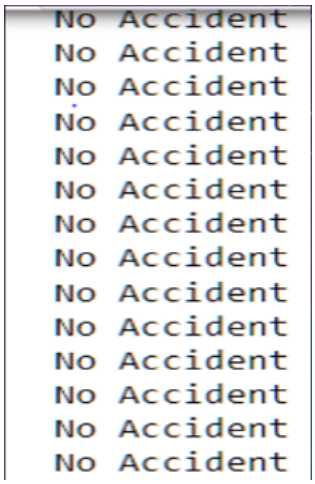
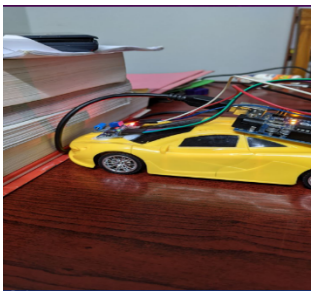


Figure 16. Car is idle or moving without an accident.



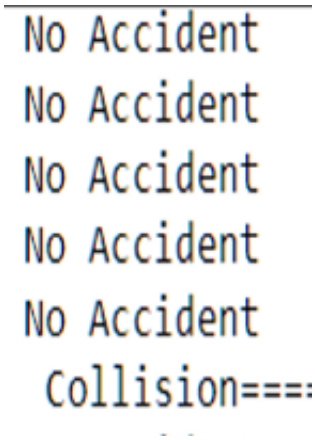


Figure 17. Car goes through a collision while moving.



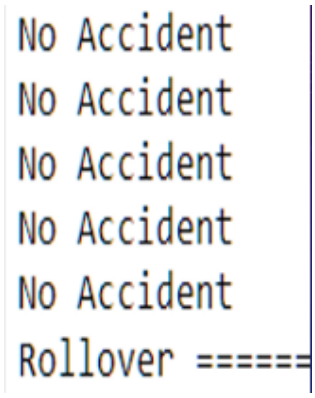


Figure 18. Car has rolled over due to an accident.

From system and real-life testing, the accuracy of detecting accidents is high for the KNN algorithm. The algorithm effectively classifies the type of accident within an instant. The system can run constantly and reliably while connecting to the cloud service, where it is ready to send accident details immediately if the rider meets with an accident. The system is cost-efficient and compact, so it can fit inside a motorcycle seat and become accessible to many users. With the system working as intended, the time taken for a person to reach hospital after a crash can be drastically reduced, increasing the survival chance of accident victims.

Future study will focus on determining the most appropriate categorization version of the ADC system, enforcing validation techniques such as K-fold cross-validation, and analyzing algorithmic time requirements. Since incident numbers may increase, rapid access to specific event facts can help first responders respond more effectively and anticipate the situations they will encounter.

REFERENCES

[1] C. V. S. Babu, N. S. Akshayah, and R. Janapriyan, "Iot-based smart accident detection and alert system," in *Handbook of Research on Deep Learning Techniques for Cloud-Based Industrial IoT*. IGI Global, 2023, pp. 322–337.

[2] N. Kumar, D. Acharya, and D. Lohani, "An iot-based vehicle accident detection and classification system using sensor fusion," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 869–880, 2020.

[3] H. M. Sherif, M. A. Shedid, and S. A. Senbel, "Real time traffic accident warning system using wireless sensor network," *Journal of Data Processing*, vol. 4, no. 1, p. 29, 2014.

[4] A. Shaik, N. Bowen, J. Bole, G. Kunzi, D. Bruce, A. Abdelgawad, and K. Yelamarthi, "Smart car: An iot based accident detection system," in *2018 IEEE Global Conference on Internet of Things (GCIoT)*. IEEE, 2018, pp. 1–5.

[5] B. S. Anil, K. A. Vilas, and S. R. Jagtap, "Intelligent system for vehicular accident detection and notification," in *2014 International Conference on Communication and Signal Processing*. IEEE, 2014, pp. 1238–1240.

[6] M. A. Rakhonde, S. A. Khoje, and R. D. Komati, "Vehicle collision detection and avoidance with pollution monitoring system using iot," in *2018 IEEE Global Conference on Wireless Computing and Networking (GCWCN)*. IEEE, 2018, pp. 75–79.

[7] A. Verma, A. Gupta, D. Kaushik, and M. Garg, "Performance enhancement of iot based accident detection system by integration of edge detection," *Materials Today: Proceedings*, 2021.

[8] P. Karmokar, S. Bairagi, A. Mondal, F. N. Nur, N. N. Moon, A. Karim, and K. C. Yeo, "A novel iot based accident detection and rescue system," in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE, 2020, pp. 322–327.

[9] M. U. Ghazi, M. A. Khan Khattak, B. Shabir, A. W. Malik, and M. S. Ramzan, "Emergency message dissemination in vehicular networks: A review," *IEEE Access*, vol. 8, pp. 38 606–38 621, 2020.

[10] W. Farooq, M. A. Khan, and S. Rehman, "A novel real time framework for cluster based multicast communication in vehicular ad hoc networks," *International Journal of Distributed Sensor Networks*, vol. 12, no. 4, 2016.

[11] A. B. Faiz, A. Imteaj, and M. Chowdhury, "Smart vehicle accident detection and alarming system using a smartphone," in *International Conference on Computer and Information Engineering (ICCIE)*, 2015, pp. 66–69.

[12] M. Ozbayoglu, G. Kucukayan, and E. Dogdu, "A real-time autonomous highway accident detection model based on big data processing and computational intelligence," in *IEEE International Conference on Big Data*, 2016, pp. 1807–1813.

[13] M. Manuja, S. Kowshika, S. Narmatha, and G. W. Theresa, "Iot based automatic accident detection and rescue management in vanet," *SSRG International Journal of Computer Science and Engineering*, pp. 36–41, 2019.

[14] A. A. Elngar, B. Thiyaneswaran, K. Anguraj, S. Kumarganesh, K. M. Sagayam, and Muthmainnah, "Iot based smart cold chain temperature monitoring with alert system for vaccination container directed to universities and schools," in *UKI Toraja International Conference of Education and Science*, 2021.

[15] B. Thiyaneswaran, E. Ganasri, A. H. Hariharasudan, S. Kumarganesh, K. M. Sagayam, H. Dang, and A. Alkhayyat, "Development of laser beam cutting edge technology and iot based race car lapse time computational system," in *Low Power Architectures for IoT Applications*. Springer, 2023, pp. 173–187.

[16] P. Srinivasan, S. Anthoniraj, K. Anguraj, S. Kumarganesh, and B. Thiyaneswaran, "Development of keyless biometric authenticated vehicles ignition system," *Materials Today: Proceedings*, vol. 81, no. 2, pp. 464–469, 2021.