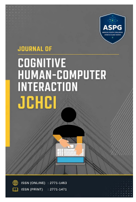
Journal of Cognitive Human-Computer Interaction

♦ **ISSN:** 2771-1463 **Vol:** 09, No. 01, 2025 ♦ PP. 13–19

Article DOI: https://doi.org/10.54216/JCHCI.090103



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 24, 2024 Revised: December 31, 2024 Accepted: January 27, 2025 ⋆ Corresponding author

ABSTRACT

Road accidents and emergency-services delay are significant issues. To overcome these issues, efficient handling of accidents through immediate detection and timely aid is crucial. Accident detection and emergency systems based on the Internet of Things (IoT), with minimum delay, are gaining significant attention in industry and academic literature. Several researchers have investigated IoT technology to detect accidents. In this work, we propose an effective accident-detection method by employing five sensors not only to detect an accident but also to report the type of accident, such as collision, no accident, rollover, or fall-off. In addition, the status of the accident is communicated to the IBM Watson Cloud platform. The incoming data received in the Node-RED platform is integrated with Google Maps to show the location and other information about the accident, which can be accessed by the hospital through a website, while alert messages are sent to victim acquaintances. In addition, two machine-learning (ML) models based on K-Nearest Neighbor (KNN) and Naïve Bayes (NB) are compared to find the best accident-detection model. It is noticed that the KNN model is the most effective ML model, which is employed to identify accident status and enhance the system by providing patient details, a kill switch, and repeated messages until acknowledgement is received.

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

1. INTRODUCTION

Nowadays, the number of accidents occurring in the world rises with the increase in population and number of vehicles. A recent World Health Organisation (WHO) 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 emergency medical-care response time may lessen the likelihood of death.

IoT is one of the most promising information and communication technologies used to reduce accident response time. It possesses an interconnected network of various physical components and devices to track and control automobiles. Significant research has been carried out to detect accidents and reduce response time after accidents [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 via sensor fusion. Three distinct 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. Its highest accuracy was observed for collisions, with an F1-score of 0.97, followed by rollover, no-accident, and fall-off occurrences.

Sherif et al. [3] reported a real-time traffic accident warning system employing wireless sensor networks (RTTAWS) and Radio-Frequency Identification (RFID) technologies. Three algorithms are employed to detect accidents, including sensor status, RFID reader location, vehicle number, number of passengers, and periodic vehicle location.

Shaik et al. [4] proposed automatic notification and response to the scene using IoT. Accelerometer and GPS sensor signals are automatically transferred to the cloud, and from there subscribers to the car receive an alarm message. The signal identifies collision severity and GPS location, allowing an ambulance to rapidly arrive at the scene.

Anil et al. [5] used flex sensors and an accelerometer to identify accidents and send nearby hospital-location information. The GPS module provides the location, which the GSM modem sends as a message. The approach also claims to broadcast live footage to convey issue severity.

Rakhonde et al. [6] reported a system that implements accident detection, accident prevention, and pollution detection. The accident is detected using an accelerometer sensor and forwarded to a node microcontroller, which then checks whether the driver is okay to prevent false alarms. If the driver does not answer, the server receives a message with the GPS module location using the MQTT protocol, and concerned persons receive the accident location.

Verma et al. [7] compared frame size and time consumption in accident detection. They also eliminated unnecessary graphical content and lightened images to make the method more portable and enable fast processing. Other works have proposed IoT-based accident-detection and rescue mechanisms that provide exact accident location and help ambulances take the shortest path to the accident spot, rescuing victims in the least possible time [8, 9, 10]. VANET-based approaches treat each moving vehicle as a node and communicate alert messages through RF modules when accidents occur [11]. Smartphone-based systems use Android applications to identify auto crashes, measuring speed and tilt angle using an accelerometer sensor and GPS unit [12]. Several computational models, including nearest neighbor, neural networks, and regression trees, have also been used for collision-detection models [13].

The contributions of this work are as follows:

- Receive inputs from multiple sensors.
- Feed the information into the selected model to examine whether an accident has happened.
- Immediately share the location with the closest hospital to dispatch an ambulance.
- Report not only accident occurrence but also accident type, such as crash, rollover, or 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 from the Covid-19 pandemic [11]. Though India accounts for only around 1% of the world's vehicles, it makes up 11% of global deaths due to road accidents according to a World Bank report. Out of 450,000 accidents annually, 33.3% of people die. A person dies every four minutes from road accidents, as stated by the same report.

Doctors specializing in trauma identify a golden period within the first 60 minutes after an accident. The care administered to the patient and the successive transport of the patient to the nearest hospital can make all the difference in the outcome. Evidence confirms that road-traffic-accident victims have more favourable results when the pre-hospital procedure is performed efficiently [12]. Hence, the proposed method aims to minimize this gap as much as possible.

3.1 Detection System

The accident-detection system implementation begins with three sensors. These sensors are chosen specifically to meet requirements. The MPU6050 is a single sensor that can input current accelerometer and gyroscope values simultaneously, as shown in Figure 1. This reduces size and cost. The vibration sensor is used on the four sides of the vehicle. A vibration sensor is chosen instead of a switch-crash sensor because a vehicle surface is large and the vibration sensor covers a larger region than a traditional crash sensor, which could miss accidents if it does not sense impact at a particular region. A flame sensor is used to detect fire, as fire and accidents can occur together.

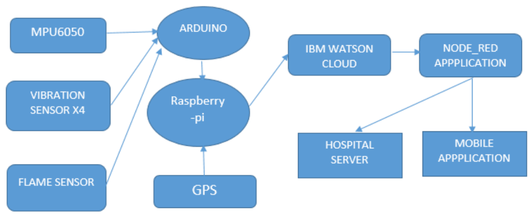


Figure 1. Accident detection system.

All these values are taken as inputs by the Arduino Uno microcontroller and serially transmitted using the serial monitor to the Raspberry Pi 3. The Raspberry Pi cannot take inputs directly because of possible overload. The Raspberry Pi also reads the current GPS position and is ready to send the location. On the Raspberry Pi, several machine-learning models are performed, their accuracy is analysed, and the best model is selected based on accuracy and ability to classify several accident classes.

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, the information is transferred to the Node-RED application, which alerts concerned 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 vibration, MPU6050, and flame sensors.

3.2.1 MPU6050

The MPU6050 sensor measures two properties. The three axes required for the gyroscope and accelerometer are recorded. Only the Vcc, Gnd, Serial Clock, and Serial Data

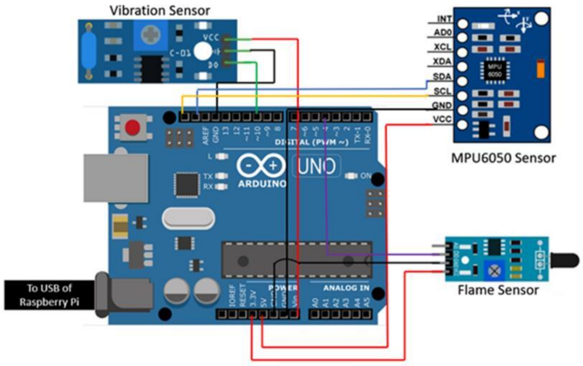


Figure 2. Arduino UNO interface with sensors.

pins are needed. The connection is shown in Figure 3.

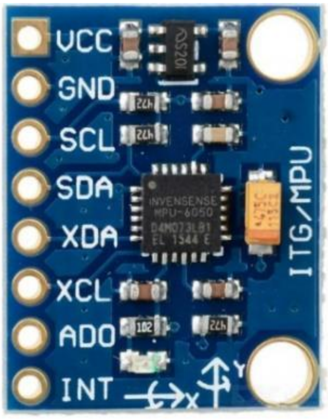


Figure 3. MPU6050 sensor.

3.2.2 Accelerometer

If a car becomes unstable due to an accident, it may no longer be in a stable position. It could swerve such that its orientation changes. This can be picked up by the accelerometer, as the car will no longer stay parallel to the gravitational axes [11]. When an accident occurs, the car automatically decelerates and impact is distributed along the three axes. Absolute linear acceleration (ALA) is calculated as

$$\mathrm{ALA} = \sqrt{A_X^2 + A_Y^2 + A_Z^2}. \tag{1}$$

3.2.3 Gyroscope

Using the gyroscope, pitch, roll, and yaw values are obtained. However, only roll and pitch values are considered. It was found that when any one of these values is higher than 90 degrees, the car is likely experiencing a rollover. Accelerometer values coupled with gyroscope values provide the required data about car orientation.

3.2.4 Vibration Sensor

The vibration sensor shown in Figure 4 is employed to identify any impact or collision that the car experiences [13]. Sensor sensitivity was adjusted so that it detects a change only when the car is hit hard on a surface.

3.2.5 Flame Sensor

When a crash occurs, because of gas presence and susceptibility to sparking, fires may be anticipated. In the event

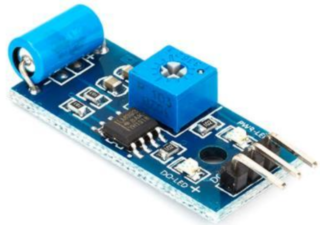


Figure 4. Vibration sensor.

of fire, paramedics must be prepared to deal with related injuries, including respiratory problems and physical burns [14]. Firefighters can also be called if needed. The flame sensor structure is shown in Figure 5.

Figure 5. Flame sensor.

3.2.6 Pressure Sensor

The pressure sensor is included to calculate altitude change if the vehicle falls from a high cliff or bridge. With atmospheric pressure assistance, the change in height can be calculated.

3.2.7 Arduino UNO

The Arduino UNO, shown in Figure 6, is a handy microcontroller that collects all required data and communicates these details to the Raspberry Pi, which performs the heavier processing for machine-learning models [15]. The Arduino has ample Vcc pins and digital I/O pins to accommodate all sensors.

Figure 6. Arduino Uno.

3.2.8 Raspberry Pi

Raspberry Pi 3B+, shown in Figure 7, is used to process sensor data as well as GPS/GSM data because it can handle complex software applications such as machine learning. Additionally, the Raspberry Pi communicates accident location, driver details, and other accident-related information to IBM Watson Cloud [16]. These details are transmitted using cloud-platform features.

Many applications are made with Raspberry Pi, but it is not strongly recommended for high-end ML applications because it lacks proper hardware for heavy computational processes and has limited RAM and processor speed. Hence, a trained neural network is used on the Raspberry Pi.

Figure 7. Raspberry Pi.

3.2.9 GPS

The GPS sensor shown in Figure 8 gives inputs in the form of NMEA messages. These messages collate longitude and latitude values and other data such as speed and altitude. In this instance, the pressure sensor that was previously planned is not required.

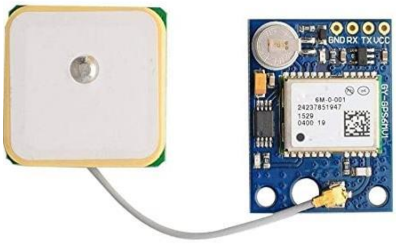


Figure 8. GPS.

4. IMPLEMENTATION OF ACCIDENT DETECTION DEPENDS ON IOT

4.1 Collecting the Dataset

For dataset gathering, individual properties come from the various Arduino ports. Accelerometer values from the MPU6050 are normalized using an equation to ensure that the model remains relevant if data changes with respect to car location. Gyroscope values, also obtained from the MPU6050, measure pitch, roll, and yaw and convey the direction in which the car is swaying, which helps determine whether rollover has taken place. The vibration sensor acts as the impact sensor, identifying collision. There are currently three outputs for the machine-learning model: no accident, collision, and rollover. The outputs represent 0—No accident, 1—Collision, and 2—Roll Off. Using the Arduino, these values are sent to the serial monitor, from which data is streamed into Excel using the data-streamer tool. They are then randomized to ensure the model is trained well. Samples of the collected dataset are shown in Tables 1 and 2.

Table 1. Dataset collection with outputs: No Accident and Collision.

AX	AY	AZ	ALA	Temp.	GX	GY	GZ	Vib.	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	Temp.	GX	GY	GZ	Vib.	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 and Other Algorithms

While creating the dataset, accelerometer values are too random for the machine to understand because there are three axes. Therefore, a normalization tool is needed. A method using acceleration magnitude instead of directions is adopted, leading to Applied Linear Acceleration. This makes it easier for the machine to understand when a vehicle meets with an accident.

The collected data is processed using KNN and Naïve Bayes. These models classify sensor readings into accident categories. The objective is to identify whether the data corresponds to no accident, collision, or rollover, while minimizing false positives and enabling rapid cloud communication.

The connection between Raspberry Pi and the hardware system is shown in Figure 9. Once the system is connected, values are transferred to the cloud through IBM Watson and viewed in recent events, as shown in Figures 10 and 11.

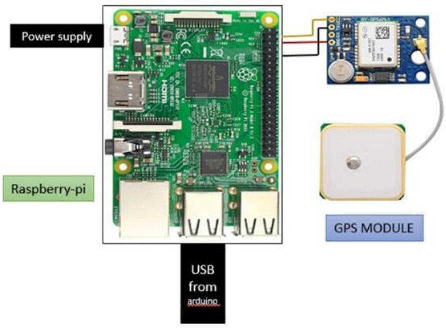


Figure 9. Raspberry Pi connected to the accident-detection hardware.

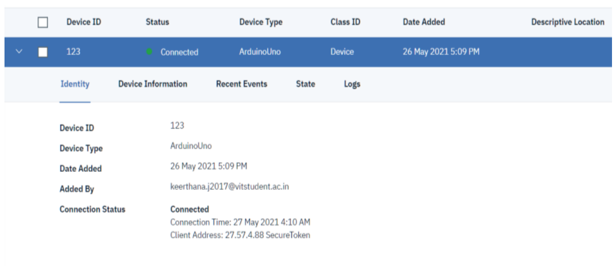


Figure 10. Connection made between Pi and Cloud.

After values are conveyed from the device, the information is used by creating a standard application and recording API-key and related details. Next, a basic website is created using the Node-RED application in the IBM Watson Platform. After entering the Node-RED flow editor, the flow is created and deployed to obtain the website. Figure 12 shows the flow used for the website.

The values received in JSON format are extracted into individual properties and compared with thresholds. Then, the

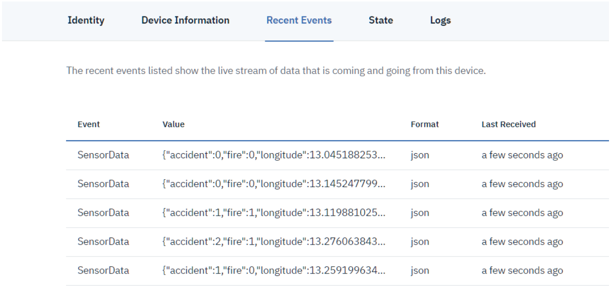


Figure 11. Values being received in cloud.

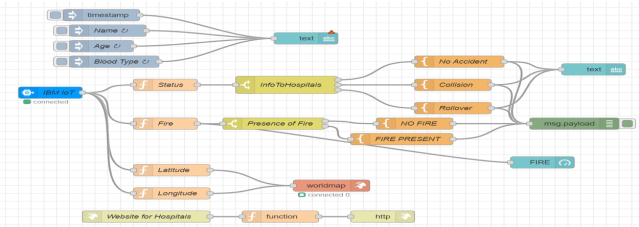


Figure 12. Node-RED flow for using data and sending it to the hospital server.

respective message is sent to the hospital and loved ones. The details can be viewed on the web page, and the position on the map is shown using UI features and the world-map feature, as shown in Figure 13.

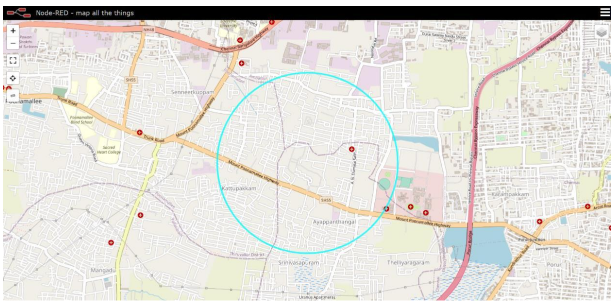


Figure 13. Node-RED map image.

Fast2SMS is an SMS service provider used for sending emergency messages depending on whether an accident happened. It is activated using a simple authorization key and token to get messages on the phone, as shown in Figure 14.

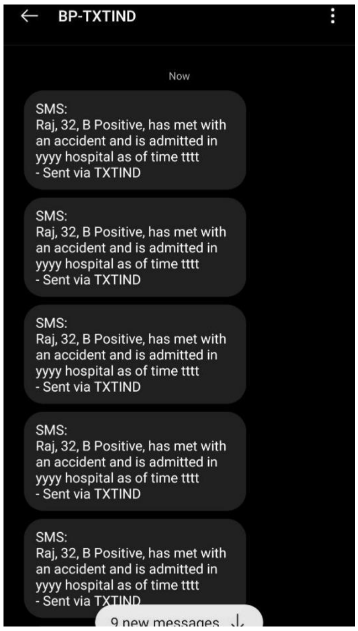


Figure 14. Emergency message to mobile phone.

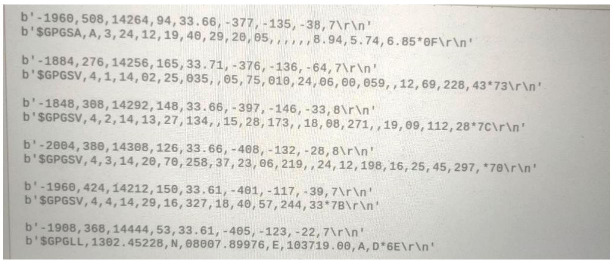


Figure 15. Output of Arduino, transferring the sensor values to the Raspberry Pi.

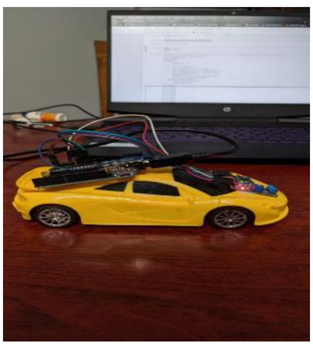
5. RESULTS AND DISCUSSION

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

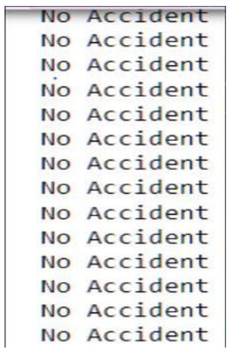
Table 3. Dataset collected using Data Streamer in Excel.

AX	AY	AZ	ALA	Temp.	GX	GY	GZ	Vib.	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 reports no accident, as shown in Figure 16.



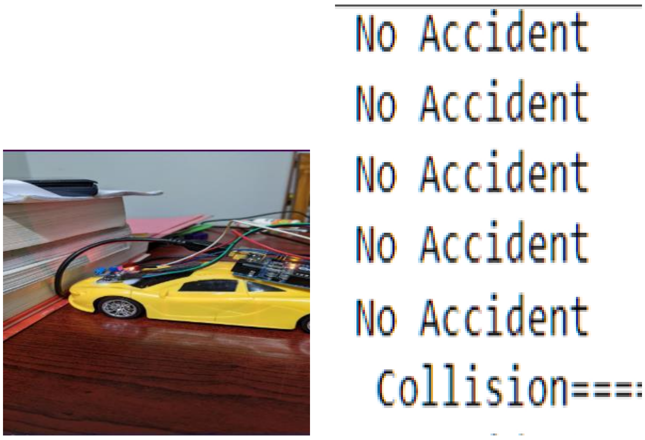
(a) Vehicle condition.



(b) System output.

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

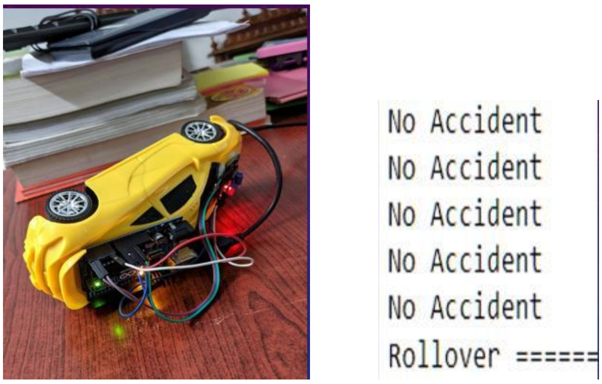
When the car goes through a collision while moving, the system reports collision, as shown in Figure 17.



(a) Vehicle condition. **(b)** System output.

Figure 17. Car goes through a collision while moving.

When the car rolls over due to an accident while moving, the system reports rollover, as shown in Figure 18.



(a) Vehicle condition. **(b)** System output.

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

6. CONCLUSION AND FUTURE SCOPE

From system and real-life testing, it is observed that accident-detection accuracy is high for the KNN algorithm. The algorithm effectively classifies accident type 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 when a rider meets with an accident. The system is cost-efficient and compact enough to fit inside a motorcycle seat, making it more accessible to many people. With this system working as intended, the time taken for a person to get to the hospital from the time of crash can be drastically reduced, increasing the survival chances of accident victims.

Future study will focus on determining the most appropriate version of the ADC system categorization, enforcing validation techniques such as K-fold cross-validation, and analysing the time required when employing an algorithm. Moreover, because incident numbers may increase in the future, first responders can respond more effectively and anticipate situations by having rapid access to facts about the event.

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 *Proc. 2018 IEEE Global Conf. Internet of Things (GCIoT)*, 2018, pp. 1–5.

[5] B. S. Anil, K. A. Vilas, and S. R. Jagtap, "Intelligent system for vehicular accident detection and notification," in *Proc. 2014 Int. Conf. Communication and Signal Processing*, 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 *Proc. 2018 IEEE Global Conf. Wireless Computing and Networking (GCWCN)*, 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 *Proc. 2020 Third Int. Conf. Smart Systems and Inventive Technology (ICSSIT)*, 2020, pp. 322–327.

[9] M. U. Ghazi, M. A. K. 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, p. 8064908, 2016.

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

[12] A. B. Faiz, A. Imteaj, and M. Chowdhury, "Smart vehicle accident detection and alarming system using a smartphone," in *Proc. Int. Conf. Computer and Information Engineering (ICCIE)*, Rajshahi, Bangladesh, 2015, pp. 66–69.

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

[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 *Proc. UKI Toraja Int. Conf. Education and Science (UKITOICES)*, 2021.

[15] M. Appalaraju, A. K. Sivaraman, R. Vincent, N. Ilakiyaselvan, M. Rajesh, and U. Maheshwari, "Machine learning-based categorization of brain tumor using image processing," in *Artificial Intelligence and Technologies*, R. R. Raje, F. Hussain, and R. J. Kannan, Eds. Singapore: Springer, 2022, vol. 806, pp. 283–292.

[16] R. U. Maheshwari, B. Paulchamy, and B. K. Pandey, "Enhancing sensing and imaging capabilities through surface plasmon resonance for deepfake image detection," *Plasmonics*, 2024.