Research Article – Vol. 02, Issue 04, No. 05

Advanced in Engineering and Intelligence Systems

Journal Web Page: https://aeis.bilijipub.com

Knowledge Management in Road Accident Detection based on Developed Deep Learning

BVSS SUBBARAO[1,*], Dr.Ramana K[2]

[1] *Department of Management Studies, Kumaraguru college of Technology, Coimbatore, India*

[2] *PG Department of Business Administration, Maris Stella College, Vijayawada, India*

Highlights

- Introduction of a novel Internet of Things-based system for accident detection using low-cost smartphone sensors.
- Recognition of the limitations in existing hardware-based systems, highlighting the need for a more cost-effective and widely available solution.
- Utilization of smartphone sensors for collision detection, addressing issues like sensor damage and failure detection.
- Development of a smartphone application to continuously read sensor data, with cloud-based processing for accident identification.
- Proposal of a scheme that not only identifies accidents but also alerts nearby hospitals and ambulances, contributing to improved emergency response systems.

Article Info

Received: 25 September 2023

Received in revised: 05 November 2023

Accepted: 30 December 2023

Available online: 31 December 2023

Keywords

Knowledge Management,
Road Accident Detection,
Internet of Things (IoT),
Deep Learning

Abstract

Business organizations and the research community try to precisely detect occurrences and assist in the case of a disaster. Most development systems are hardware-based, making them pricey and unavailable in every vehicle. A vehicle's sensors can be destroyed in various ways, including through minor accidents or fixed interactions. In some instances, the sensors are incapable of detecting an accident. Intelligent phone sensors are a great alternative because of their dependability and availability. Smartphone sensors can detect collisions. Few methods detect failures using cell phones. These systems, however, have a low error rate. The study proposes an Internet of Things-based system built on low-cost devices. The suggested system has two stages: identification and reporting of accidents. These systems rely on sensors to detect mobile phone failures. The suggested system employs a variety of smartphone sensors. The study involves creating a smartphone application that continually reads sensor data and sends it to the cloud for further processing. The crash was discovered by threshold analysis. The critical contribution of this research is creating a scheme that alerts nearby hospitals and ambulances when an accident occurs. The system will have more minor inaccuracies, precisely identify accidents, and perform better than earlier techniques using four sensory inputs. This paper introduces novel types of deep learning for accident detection.

Nomenclature

Indices		*Variables*	
CS	*Cuckoo Search*	a	*Price factor*
CP	*Customer Payment*	b	*Price factor*
DG	*Distributed Generation*	B_{ij}	*Susceptance of Line ij*

* Corresponding Author: BVSS SUBBARAO

Email: bvsss_rao@yahoo.co.in

DISCO	*Distribution Company*	c	*Price Factor*
GENCO	*Generation Company*	Q_D	*Demand Reactive power*
GOA	*Grasshopper Optimization Algorithm*	Q_G	*Generator Reactive Power*
LMP	*Local Margin Price*	G_{ij}	*Conductance of Line ij*
OPF	*Optimal Power Flow*	S_{ji}^{max}	*Maximum mixed power limit*
SCL	*System Cost Index*	V_i^{max}	*Upper limit of voltage at bus i*
VSI	*Voltage stability index*	V_i^{min}	*Lower limit of voltage at bus i*
Parameters		λ	*Energy marginal section in reference bus*
C	*Cost function*	$\lambda_{L,i}$	*Section associated with losses*
B	*Benefit factor*	$\lambda_{C,i}$	*Section associated with congestion*
P_D	*Demand power*		
P_{DG}	*Distributed generation power*		
P_G	*Generator power*		
∂_j	*Angle of the voltage of i^{th} bus*		
v_j	*Voltage of i^{th} bus*		

1. Introduction

Cities are becoming increasingly congested with tourists, residents, and cars. The increase in the proportion of automobiles has led to a rise in traffic. According to the last WHO estimate, 1.35 million people die yearly in traffic accidents, and 50 million are wounded [1]. Road crashes are now the ninth leading cause of death. However, the International Road Safety Association (ASIRT) predicts that, unless substantial improvements occur, they will soon increase to the fifth leading cause of death [2]. Furthermore, the societal costs of road accidents are significant. The International Road Safety Association expects that road accidents will consume one to two percent of each country's yearly budget. [2].

Even in sophisticated nations with road safety measures, the global yearly number of driving deaths has recently climbed [3]. However, the issue remains that low- and middle-income nations have the most incredible pressure on tolls and road injuries [4]. High fatality rates are particularly significant for improving road safety in developed and developing nations. The introduction of IoT promises to produce intelligent traffic control systems [5].

The Global Positioning System (GPS) is rapidly being employed in various applications, including vehicle positioning. Indeed, many automobiles nowadays have GPS systems that detect the vehicle's location and report it to cloud servers [6]. Other instruments for accident detection or smart transportation management are also integrated into current automobiles and continuously gather and store data. [7]. Because of the inclination to enhance the accuracy and efficacy of the algorithm, the high sampling rate creates major obstacles to storing and evaluating this data.

The communication of numerous items using Internet channels is called the Internet of Things [5]. In general, the Internet of Things is developed and exhibited by connecting all items over the Internet with the goal of remote sensing and control. Other technologies influencing IoT include wireless sensor networks, machine-to-machine communications, robotics, wireless networks, Internet technologies, and smart gadgets. The Internet of Things (IoT) refers to items and properties that are related and traceable via digital networks. This network is expanding daily. Businesses must alter how data is evaluated to gather, clean, prepare, and analyze RFID sensor data and tags in the least amount of time to reap the benefits of IoT.

In today's highly competitive world, deciding on the correct data in the shortest time is critical to keeping a firm running. The basic IoT architecture in Figs. 1 and 2 depicts the broad IoT ecosystem, which comprises a range of daily products.

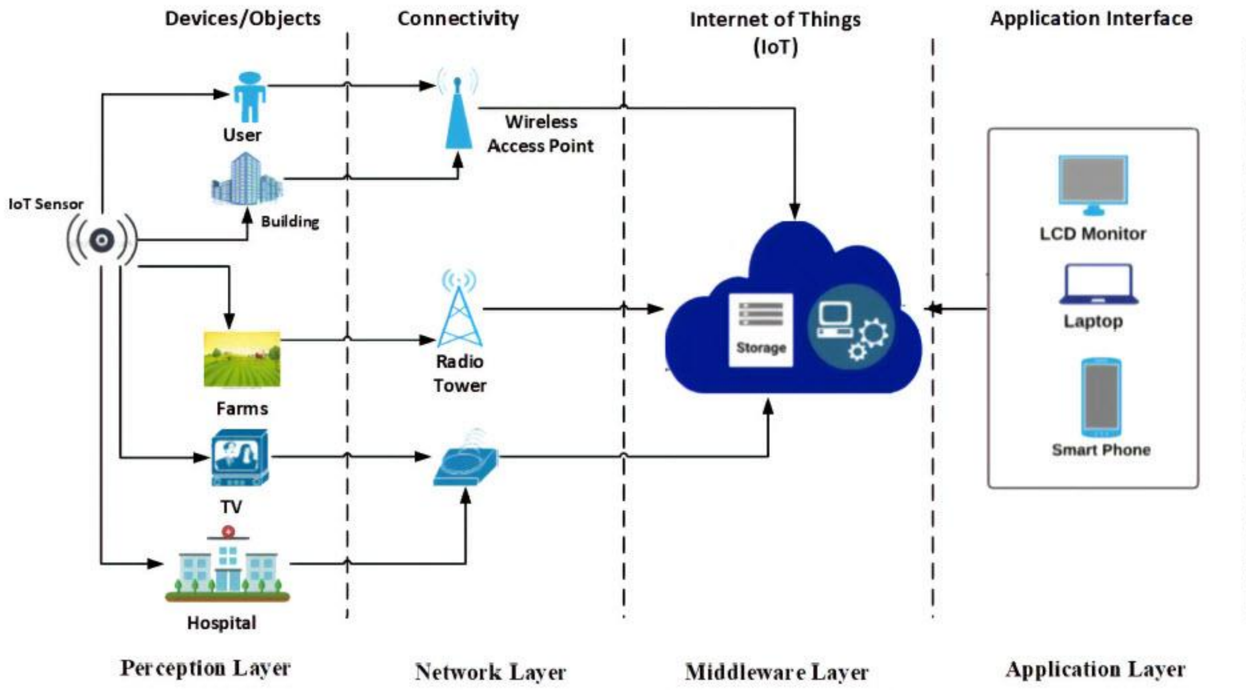

Fig. 1. IoT base architecture

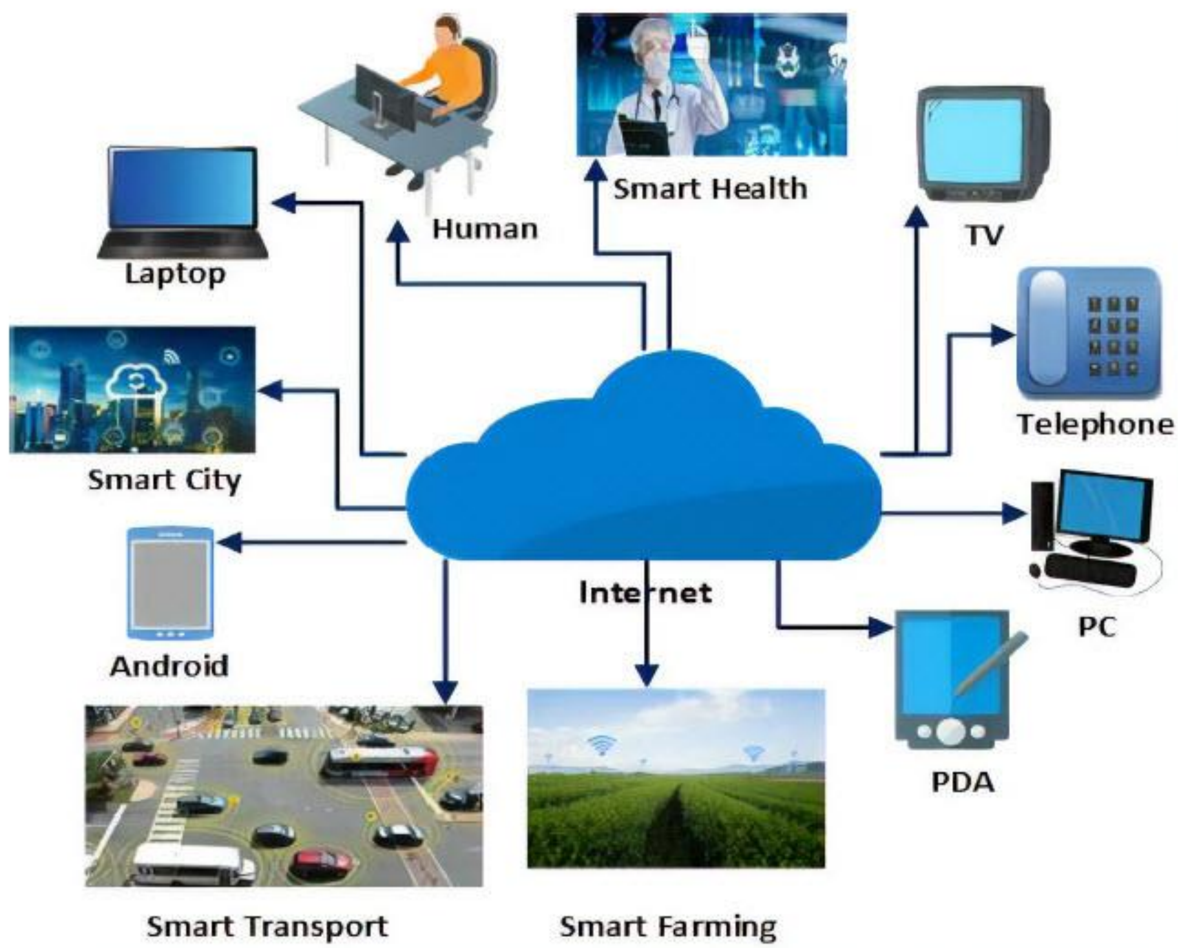

Fig. 2. IoT general ecosystem include a series of everyday objects

There are several meanings of IoT. The Institute of Electrical and Electronics Engineers, for example, defines the Internet of Things as "a network of products—each with embedded sensors" linked to the Internet. According to Internet meanings, things are a physical, cyber system that connects physical items to the cyber domain. IoT has a surprising breadth and application, including many items such as automobiles, buildings, mobile phones, different electronics and infrastructure, and even clothes. A network often connects devices in an IoT system. These material things may be RFID-enabled or contain other types of identification, such as barcodes, and various sensors may detect their existence. As input, these sensors collect information about the item and transfer it to the processing and analysis system. It should be highlighted that sensors have restricted computing power and storage size, which might pose problems, particularly regarding security and dependability. Some of these challenges have been addressed using cloud computing [8].

The quantity and diversity of sophisticated applications that use cutting-edge technology are increasing. Reduced storage prices, processing power, device availability, and affordability are critical considerations. Intelligent cities offer residents better, more innovative, responsive, and more economical services. Through evolving intelligent transportation technologies, smart cities may supply mobility solutions. Several nations use traffic systems to improve protection and minimize contamination. Recent World Bank research found that "welfare gains comparable to 6 to 32 percent of GDP can cut by 50 percent road fatalities and injuries over 24 years [8]." The primary emphasis in intelligent transportation systems is on real-time information and decision-making.

Reduced accident reaction time is one of the most effective strategies to minimize road deaths. Several systems, like e-Notify, can assist in identifying and reporting traffic accidents [8]. Each vehicle must have an onboard unit, or OBU, to use the e-Notify system. While this is a viable option, the European Commission created the eCall system and mandated its use in all vehicles manufactured. When an accident is detected, the eCall system calls 112 (999 in the UK and other countries) [8].

The main innovation of this research is the optimization of the approach presented in reference [8], which is the basic research paper of this research. In fact, the approach presented in [8] has created an innovative environment that works in the web context, but the purpose of this research is to provide a simulation structure for the IoT platform for identifying and reporting accidents in smart cities and evaluate it. Some challenges of IoT and also accident detection models were reviewed in [9]–[22]. The main contribution is also using deep learning in a new mode developed with modeling based on problem issues.

The rest of this paper can be categorized as follows: In Section 2, the literature review is done. The proposed method is presented in Section 3. In Section 4, a discussion of the results is expressed, and the conclusion is also stated in Section 5.

2. Review of Literature

Several concepts and tactics for dealing with road safety, vehicle communications, and post-accident rescue operations may be found in the literature. This study concentrates on the most practical approaches and strategies: software and hardware-based solutions. It primarily focuses on accident detection technology that makes use of a variety of sensory inputs. The review of the current systems connected to traffic dangers and road accidents in this part reveals their strengths, shortcomings, and limits.

Part I) Smartphone-based systems

A substantial amount of research on this topic can be found to solve the challenge of low-cost, resilient solutions to detect and notify automobile accidents based on mobile technology. It debuted a crash alarm system in [23] that used accelerometers and GPS data from mobile devices to identify accidents. The system delays transmitting incident-related alerts. [24] proposed an accident identification approach that uses the vehicle's position and reports an accident through text message on a cell phone. The system only employed one sensor, the vehicle's location, which might lead to false accident reports. The authors of [25] presented a technique for detecting accidents that employs gravitational force, velocity, and noise. A web server receives an emergency notification and sends a message to the emergency contact number. The key downside of this technology is that it may report inaccurately in a low-speed collision where the system attempts to ensure the user is in the car.

They created an Android application that distinguishes events using acceleration data [26]. The technology transmits a pre-recorded audio message to 108 ambulances (an emergency service provided in India). The accelerometer is highlighted in [27] as the primary sensor of the smartphone for detecting an accident. The system continually collects data and utilizes that information to calculate the accident's severity. The center notifies the medical provider of the accident and gives the owner or driver information. The issue with both methods is that their dependence on a single sensor leads to mistakes in reporting because no other information is available to validate a suspected occurrence.

In [28], an innovative phone-based system is presented that uses the accelerometer to identify the accident and locate the nearest emergency location to which the occurrence should be reported. Again, this method has a break point issue, which leads to a tendency for erroneous reporting.

They created a smartphone application in [29] utilizing a processing unit, or OBU. This software allowed the driver to communicate with their automobile. The app notices an accident using airbag agents and alerts the emergency service provider through email or SMS. This software's flaw is that it needs a smartphone app. [30] also presented a smartphone-based car accident alarm system. The system identified the accident using a pressure sensor on the phone. They monitored speed using GPS and an angle gradient with a smartphone accelerometer. The system detected the mishap on a smartphone using two GPS sensors and an accelerometer. Event data is saved on the server. The system was more dependable than others, but if the server went down, it might cause issues. After detecting the incident, the system informed the nearest hospital and police station. [31] presented a mechanism for detecting accidents using smartphone sensors.

This phone sent incident information via its 3G technology connection.

In [32], a novel approach is described that uses accelerometers to detect collisions and warns the server of emergency dispatches and essential information via the Global Messaging System for Mobile (GSM). Again, this system detected an accident with only one sensor.

[33] presented a GPS and GSM-based car accident detection and tracking system. The switch buttons detect the event and use GSM to send the location to the phone number provided by the user.

Another method proposed in [34] is a GPS-based automated localization system. This method also provided communication capabilities via the use of a GSM modem. The technology used an accelerometer to detect the incident and send a notification to police headquarters and the rescue team. In the event of a minor collision, the driver has the option to disable the warning message.

[35] presented a system that uses smartphones to exchange vehicle position and speed data with other cars in real-time. Various machine learning algorithms are employed to evaluate data, offer information on road conditions, and detect accidents. The system attempted to make judgments based on the information gathered. Unfortunately, the findings of this investigation show inadequate accuracy in accident detection.

[36] presents yet another well-respected study. Using an integrated accelerometer and gyroscope, the writers developed a smartphone-based application that detects unintended accidents and alerts the adjacent emergency. The planned strategy is focused on reducing reaction time and does not consider vehicle accidents.

Part II) Hardware-based Systems

As mentioned, road accidents are the primary cause of fatalities; thus, more study is needed to discover and swiftly begin rescue efforts. The likelihood of fatality is lowered if the timeframe between the crash and dispatch of the rescue team is shortened, and this has inspired several researchers to minimize reaction time. [37] presented a study of mobile phone appliances and events using OnStar data. The On Star system has an implanted mobile card that may be activated manually or automatically via airbag deployment. The technology has a restriction in that it requires manual intervention or is triggered in the case of a catastrophic accident in which the airbag is deployed and depends on a sensor signaling that the airbag must be organized.

Some systems only run in specific regions or employ exceptional instances. Because there are multiple competing movements in various directions, road junctions are a typical location for crashes. [38] presented a system for automatically detecting, recording, and reporting incidents at junctions. Cameras are installed at junctions to detect vehicles and their associated data. The system's choice is based on the extracted characteristics. The factors that contributed to the incident, as well as the characteristics of road crossings that impact safety. This technology only detects accidents at junctions and does not identify accidents anywhere else. The authors presented a crash tracking system in [39], which employs a microcontroller to regulate all processes. Messages are delivered to a determined cellphone number. The system's performance review revealed a bogus report of accidents. The current system is simply for accident detection and has no connection to rescue systems.

[40] describes an intelligent system capable of detecting an accident and utilizing emergency service information. Its intensity is utilized in choosing whether or not to employ emergency rescue resources. Delay, bandwidth, and delivery assurances are the issues this research tackles when running the system. [41] proposes a crash detection and reporting system that detects accidents using a sensor. [42] presents a solution for identifying

events that leverages backup vector machines and IoT. This technique is used for both accident detection and traffic forecasting.

A GPS-based position tracking system was presented in [43]. The system uses a crash sensor to gather event data. The data is subsequently sent to the emergency center via SMS. The response team is then dispatched to the location by the Emergency Center. Despite its many benefits, the system has several disadvantages. For example, the user must manually start the system because it is not automated. The [44] IoT-based system detects shocks with a shock sensor. The rescue crew received basic rescue information from this mechanism. This strategy aided the rescue team in determining the exact location of the accident. It also made people realize the quickest and best path, which is then relayed to the nearest ambulance.

3. Proposed Method

In order to address the current limitations in crash detection systems, this study proposes an Optimized Accident Detection and Reporting System, or OADRS. The new OADRS crash tracking and reporting system utilizes modern Android smart phone capabilities and thus reduces overall cost because no special hardware is required. All processing is done in the cloud. The OADRS architecture is the layer architecture seen in Fig. 3.

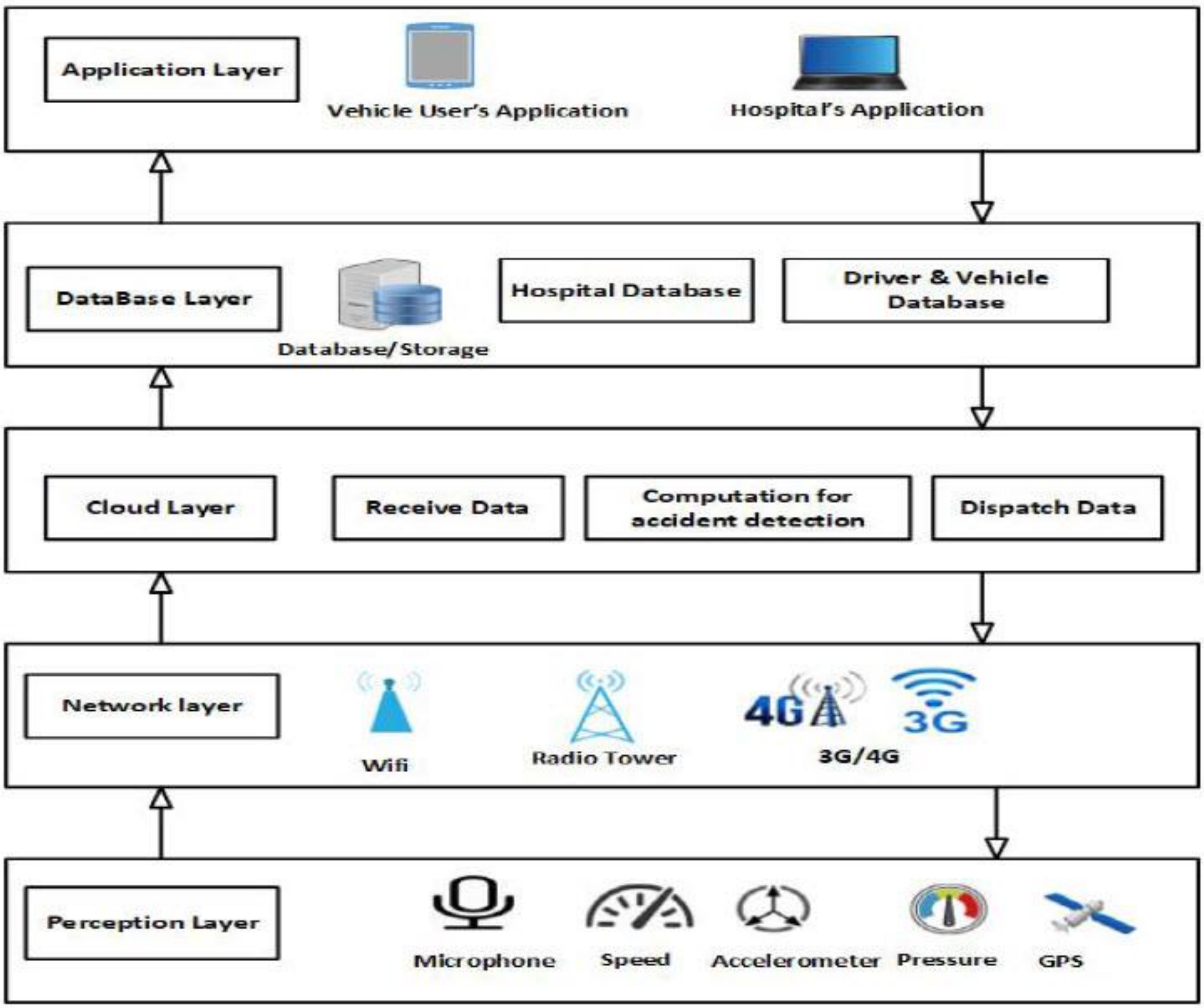

Fig. 3. OARDS architecture

The OADRS system architecture comprises five diverse coatings: the application, database, cloud, network, and perception layer. The perception layer is responsible for interacting with the smartphone sensors in the suggested architecture, and gathering data from the sensors is the primary aim of the perception layer in OADRS architecture. This data related to the vehicle's gravitational force, speed, sound, pressure, and position was obtained via smartphone sensors before being sent to the network layer for further processing. The network layer serves as a link between the perceptual and cloud levels. First, it gets data from perception layer sensors such as smartphone sensors, locations, and operator information. The data is then sent to the cloud via the network layer using Wi-Fi cellular connectivity or 3G/4G/5G technologies. The cloud layer maintains the algorithm for accident identification and detects an accident based on threshold analysis, so the nearest hospital is contacted if an accident is recognized. The data processing layer subsequently sends information to the database. Finally, the database layer keeps information on accidents, hospitals, drivers, and vehicles. The application layer receives all information, including the smartphone driver application interface and the hospital's web-based system interface.

For a better understanding of the system, Fig. 4 displays the work of the suggested method.

First, users download the app and install it on their smartphones. Then, log in to the application and provide the necessary information. Once registered, users can use the app freely. They activate the tracking process each time by initiating a trip. Then, the smartphone reads sensor data and sends it to the cloud. In the cloud, this information is processed to recognize any accident. A nearby hospital will be notified and provide details of the incident at the moment of the accident.

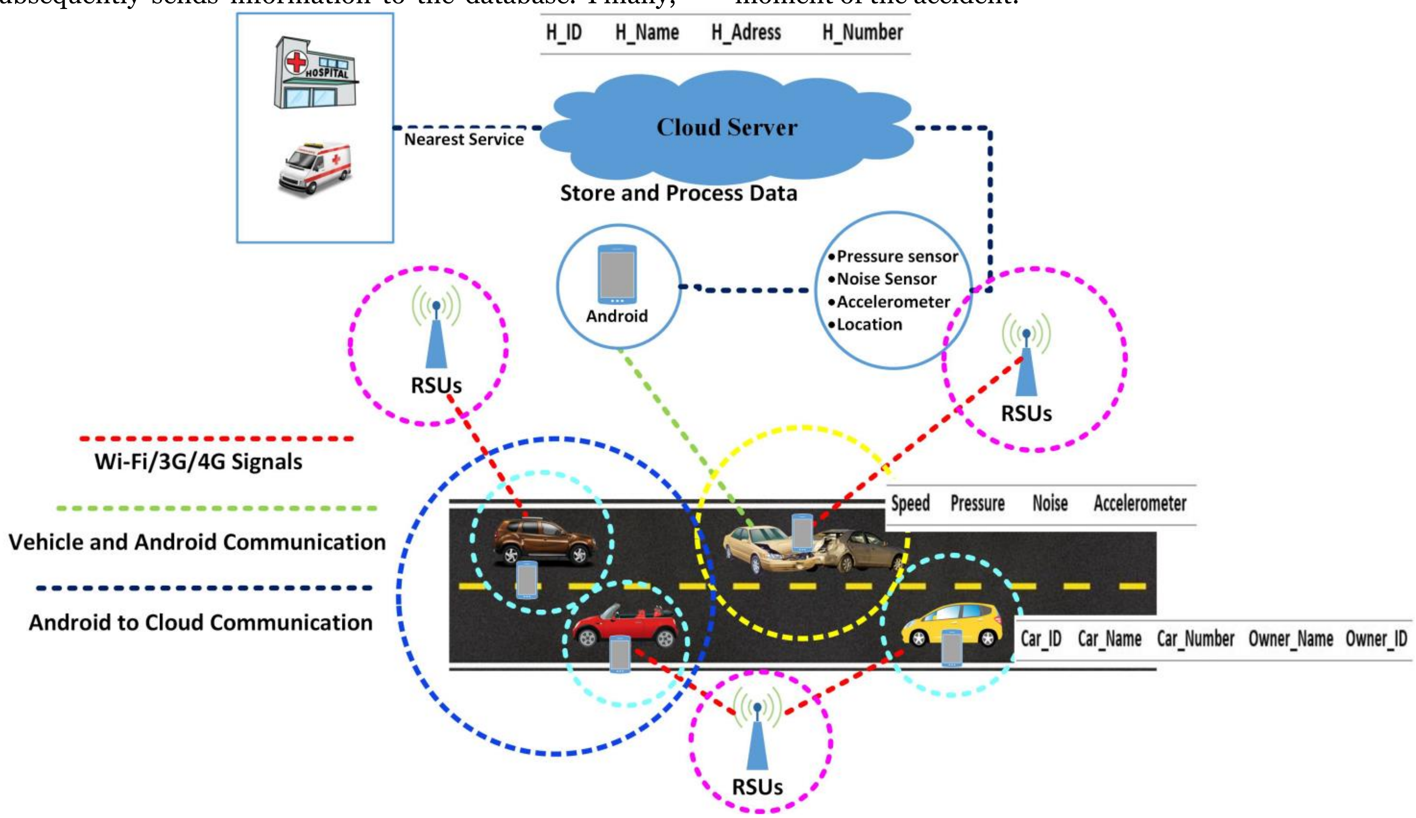

Fig. 4. OARDS flow diagram

If each automobile is linked to a smartphone, each smartphone contains four sensors: pressure, sound (microphone), accelerometer, and a speed sensor. An Android phone collects raw data which is equipped with the above sensors to use for the experimental evaluation. The phone continuously sends data to the cloud. In the presented method, roads are equipped with roadside units, or RSUs. RSUs are used to store information from cars. For testing, there are five vehicles (V1, V2, V3, V4, and V5). The V1 vehicle connects with the nearest RSU. It can be named RSU1 without losing data. In the present scenario, V2 and RSU1 communicate with each other because V2 is not in the RSU2 and RSU3 ranges. The V3 and V4 vehicles collide, so RSU1 falls into the range. Because it is out of range, crash information cannot be shared directly with RSU3, as is the case with the V2 vehicle.

The cloud analyzes data to determine whether or not an accident has occurred. There are set threshold values; if an accident happens, the sensor data offers a superior deal than the threshold value. When criteria are met, an alarm is generated and sent to the car's driver. The hospital will not be called if the motorist disregards the signal to avoid false reporting. The cloud service will alert the nearest hospital if the driver does not respond within 10 seconds. The cloud has a database of autos and hospitals. The hospital sends an ambulance to the spot for rescue operations. Hospitals also have information about ambulances. The primary purpose of this design is to increase the accuracy of accident identification. The system is separated into two stages: detection of accidents and notification. These mechanisms are discussed in further depth in the sections that follow. Fig. 5 displays a system overview.

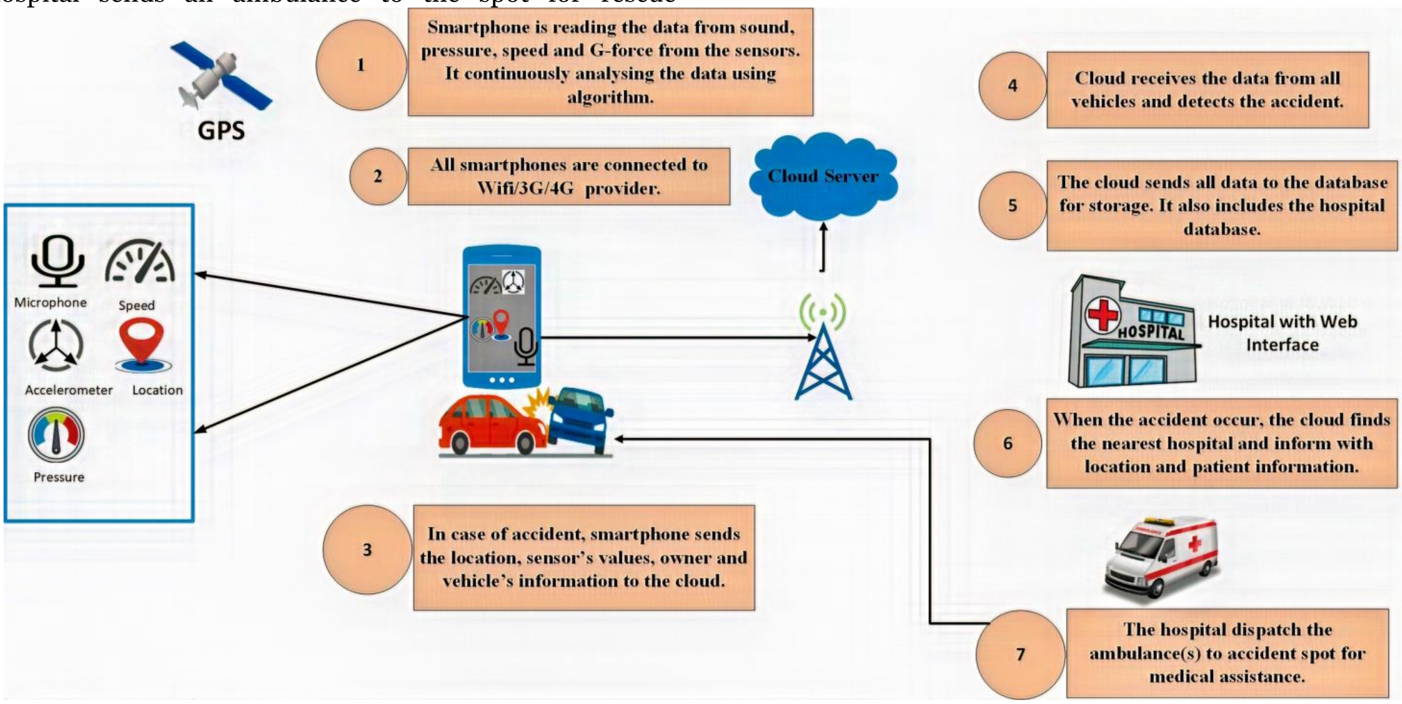

Fig. 5. Proposed approach presentation

The chief aim of the proposed system is to provide an architecture that can handle five cases: 1) direct communication of the vehicle to the infrastructure; 2) automated sharing of accident data; 3) increasing the precision of accident detection; 4) reducing fake reports and messages; 5) establishing a cost-effective system. Fig. 6 displays the activities of the presented architecture.

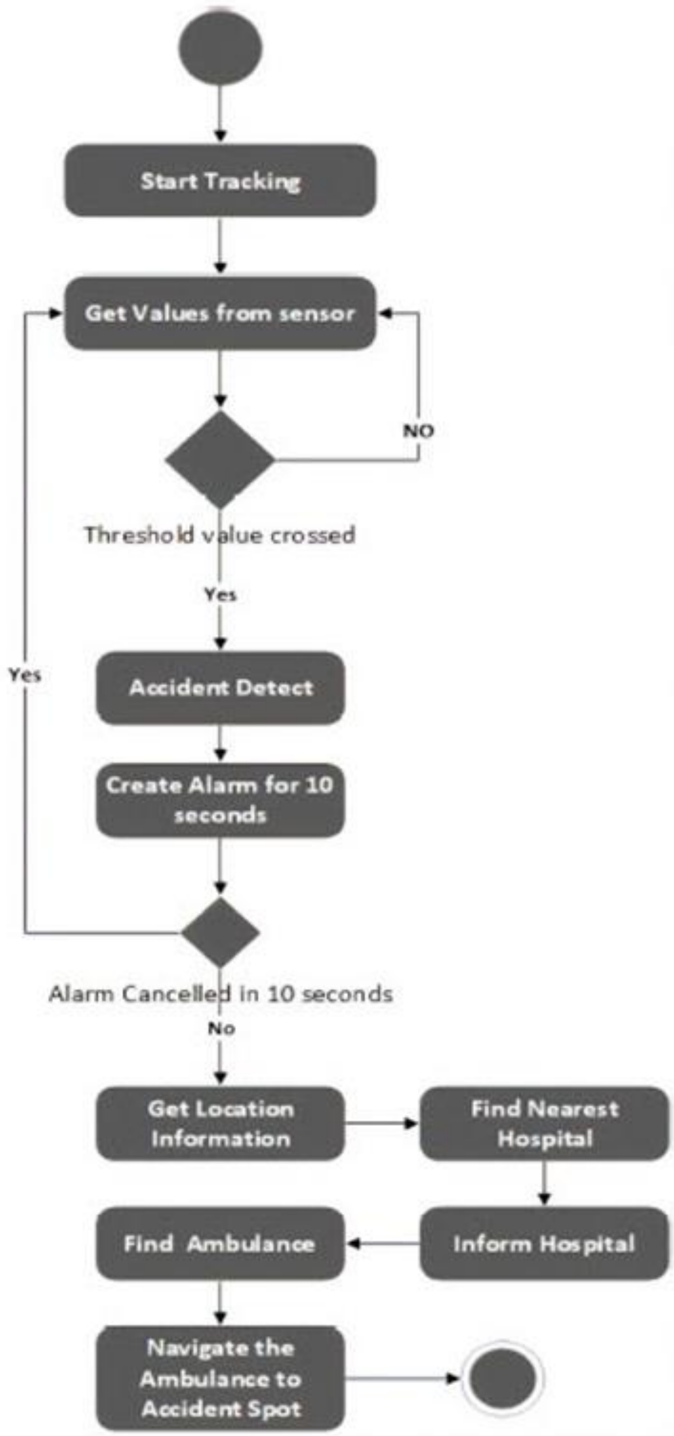

Fig. 6. Proposed approach flowchart

The most significant variables in this study include multi-smartphone sensors, including an accelerometer, GPS, pressure, and microphone to detect crashes. As well as investigating five proposed approach operations, including the possibility of direct communication of the vehicle to the infrastructure, automatic exchange of accident information, increased accuracy in accident detection, reduction of the number of false reports, and the creation of a cost-effective system, In the following, it is necessary to present a new approach in several different sections. The first part deals with the accident detection components.

Phase One: Accident Detection Components (Deep Learning Inputs):

Accident identification is used to avoid occurrences that cause damage or injury and minimize the number of people killed in traffic accidents. Fig. 7 displays the key components used during the accident detection phase. The detection process determines the presence of an accident by using data from the smartphone's microphone, accelerometer, GPS, and pressure sensor. Below is further information on the components used in the accident detection phase.

- ✓ Smartphone accelerometer sensor: This component detects accelerometer sensor data to calculate acceleration force (or G-force). One of the important components for detecting an accident is the accelerometer in a smartphone. An accident flag is triggered if the acceleration force reaches 4G. The 4G threshold number results from a mix of secondary studies and testing. Assume a car comes to a complete stop but is not involved in an accident. It is subjected to less than 1 G of force in that situation. 4 G is decided as a threshold for raising an accident flag to consider all incidents.
- ✓ GPS Technology: This component detects accelerometer sensor data to calculate acceleration force. An accident flag is triggered if the acceleration force reaches 4 G. The 4G threshold number results from a mix of secondary studies and testing. Assume a car comes to a complete stop but is not involved in an accident. In that situation, it is subjected to less than 1 G of force. 4 G is identified as a threshold for raising an accident flag, considering all incidents.
- ✓ Pressure sensor: A pressure sensor detects the pressure of an automobile in the event of a collision. This component also gathers continuous pressure data, producing an alarm if the pressure reaches a predetermined threshold of 350 Pa. The pressure sensor increases the system's accuracy and decreases the possibility of incorrect identification and accident reporting.
- ✓ Smartphone microphone: This component detects background noise. When the noise level exceeds 140 decibels, an accident flag is raised. The built-in microphone enhances accuracy and decreases the possibility of false positives. The built-in microphone detects sound.
- ✓ A built-in microphone may detect high-decibel sounds when an automobile collides. However, passengers laughing, phones falling, or loud music contribute to the noise. The sound level is set at 140 decibels. The microphone is used to increase the chance of precisely detecting an accident. The accident detection components discussed above are shown in Fig. 7.

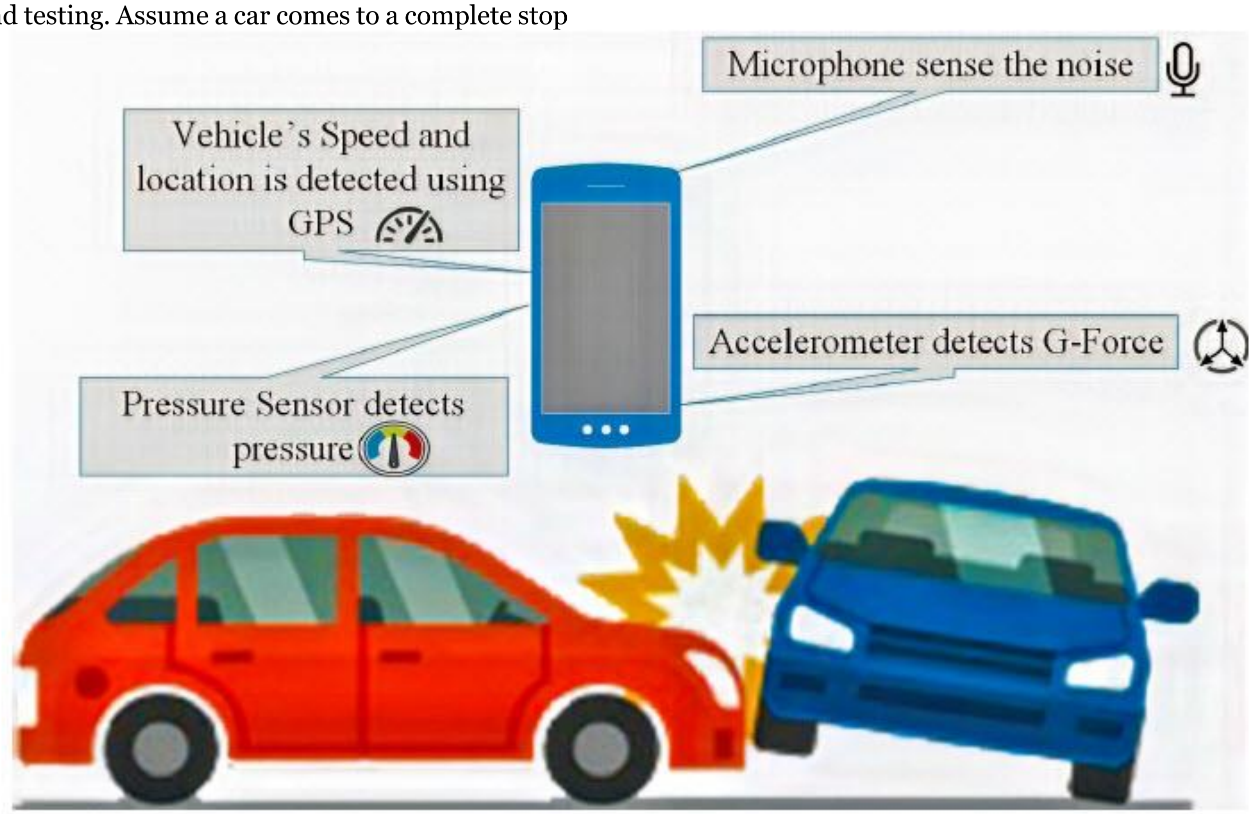

Fig. 7. Components of Accident Detection

Phase Two: Phase of The Notification

When an accident is detected, effective communication and dispatch are vital. The system locates an accident using the smartphone's GPS when it is spotted. The cloud has a hospital database and employs a mapping tool to locate the closest hospital (in our case, the Google Maps API). A message containing the location's data and owner information is sent to the hospital. The obtained data is saved in the current database. Fig. 8 depicts the execution of the notification phase.

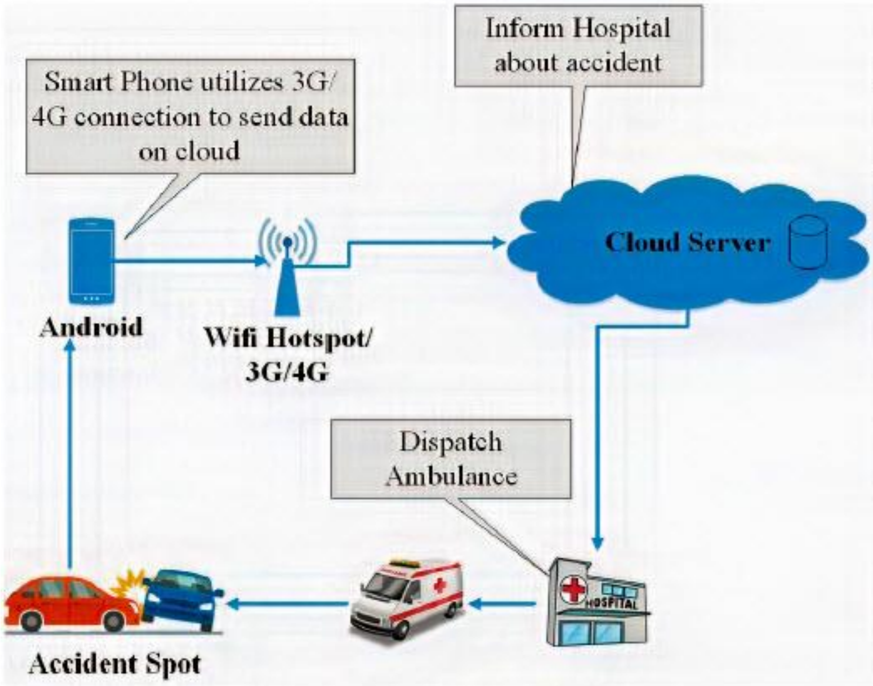

Fig. 8. The notification system components

Phase Three: Database

Database of cars: An automobile database provides all vital information regarding registered vehicles. Information about the owner, name, address, and number of automobiles is saved in the cloud to resolve any incident. The samples from the automobile database are shown in Table 1.

Table 1. Samples of cars database

ID of car	Name of car	Number of car	Name of owner	ID of owner
C 1	Suzuki	RAZ 3825	Bilal Khaled	36512-4530645-9
C 2	Landover	MP 3509	Shahid Khalid	33103-9963108-6
C 3	Toyota	LAL 76 4320	Ali Hosseini	12145-1519307-7

Database of Hospitals: The system must be aware of all adjacent hospitals to notify them of an emergency. When the system sends a message to the cloud, the cloud must locate and pass the message to the nearest hospital. Table 2 displays the information that has been saved.

Table 2. Database of Hospital

ID of hospital	Name of hospital	Address of hospital	Number of hospital
H 1	Jahan	Usmani Rd Panjab	+92-42-95231543
H 2	Health	Kohistan Rd Mumbai	+92-51-2455613
H 3	Soldierly	Abid Majeed Rd Delhi	+92-51-9260376

Sudden changes in network topology, communication, and the modifications necessary to achieve a sustainable re-establishment of topologies are the main problems with Internet of Things-based smart cities. In these networks, clustering may be used to lessen these negative effects. Cars traveling in the same direction are referred to as being in a cluster. It lessens the negative effects of recurrent clustering. This study examines the relationship between cluster size, vehicle speed, traffic, and CW size and their effects on closure losses and output efficiency. Picking the right cluster for building such networks is another problem.

As a result, using clustering to collect and transmit these requests is a beneficial way of reducing these interactions and requests. Due to the absence of direct vision and consecutive reflections created by the impact of the waves on the mountain slopes, these stations are entirely inoperable on hilly routes; if planned, they need to be utilized at each place that is not in direct view, which would result in a high cost.

The following assumptions are taken into account in the studies:

- ✓ The vehicle's relative velocity is considered.
- ✓ Elements of the hilly routes' angle of view impede a complete view.
- ✓ Errors produced by physical layer decoding are tolerated.
- ✓ Each vehicle is aware of its geographical location by implementing GPS and GIS modules.
- ✓ This procedure was used to choose the cluster based on the weighted method. This study's technique is based on the intensity of the received signal (RSS). Furthermore, the average signal strength received according to Friies' rule is recognized throughout the preceding experiments (1).

$$E(p_{re}) = \frac{P_{tr} G_{tr} G_{re} \lambda_c^2}{16\pi^2 d^2} \tag{1}$$

In this regard, the distance between the transmitter and the receiver is represented by d, as are the transmitter power P tr, G, and G re of the transmitter and reception antennas (as previously specified). (2) will be used to calculate the wavelength of the wireless signal.

$$\lambda_t = \frac{c}{f} \tag{2}$$

(2) denotes the speed of light as c and the frequency of the wireless channel as f. The US Federal Communications Commission's standards and bandwidth necessary to interact with end-of-road vehicles in 1999 heralded the beginning of a new era in inter-car communication. This standard gives inter-car communications a bandwidth of 5.9 GHz. It is defined on this frequency between seven and ten channels (5.850 GHz to 5.926 GHz), with one channel especially tailored to boost automobile safety and other channels for car-specific applications. To simplify, we'll suppose the frequency is in the 5.890 GH range. The probability density function of the received signal will be obtained as (3), given the channel assumptions.

$$F_{re}(p_{re}) = \frac{1}{E(p_{re})} e^{-\frac{P_{re}}{E(P_{re})}} \tag{3}$$

According to the IEEE 802.11P standard, an emergency message is decrypted if the received signal intensity is greater than the lowest sensitivity (P min). If this is not done, emergency communication will be lost. The receiver calculates the signal intensity when the emergency message is received. Consequently, the possibility of receiving an emergency message within the distance L tr is calculated using (4).

$$\begin{aligned} pr_{succ(d)} &= pr\{p_{re} \\ &> p\_\min | l_{ir} = d\} = 1 - pr\{p_{re} \\ &\le p\_\min | l_{ir} = d\} \\ &= 1 \\ &- \int_0^{P\_min} \frac{1}{E(p_{re})} e^{-\frac{p_{re}}{E(p_{re})}} d(p_{re}) \\ &= \exp\left[-\frac{p_{min}}{E(p_{re})}\right] \end{aligned} \tag{4}$$

If an emergency message is successfully received, the intensity of the received signal in the receiver will be defined using formula (5).

$$E(p_{re}) = -\frac{P_{min}}{inPr_{succ}(d)} \tag{5}$$

In addition, the threshold for assessing reliable communication using (1) and (5) may be determined by (6).

$$p_{tr} \ge p_{tr,th} = -\frac{16\pi^2 d^2 p_{min}}{G_{re} G_{re} \lambda_c^2 ln p_{th}} \tag{6}$$

The threshold parameter, in this case, is p_{th}. According to the preceding equation, the link between signal strength and the chance that the receiver accurately receives the message has grown with increasing distance, even when the signal intensity is small (for example, approximately dBm20). The probability that the receiver will accurately receive the message is extreme. This will become stronger and stronger when the transmitter's signal power is increased.

The excessive increase in transmitter power, on the other hand, will have an impact.

By raising the transmitter power, because of the increased interference, the number of nodes for access and posting on the channel will increase.

Another critical concern is a delay. End-to-end delays are calculated by the time it takes to send and receive messages. Delays can be classified into three types: queuing delays, computational delays, and delayed propagation. The primary delay in doing the computations is the release delay.

Even though the processing units placed in automobiles are capable, this cannot eliminate the delay caused by vehicle propagation since wireless networks' limited resources produce this delay. So, given the minor delays in queuing and processing, treat these as delayed emissions. As a result, the emission delay (T_{tr}) is the time elapsed between back-off and retransmission in the MAC layer.

The IEEE 802.11P protocol in the MAC layer may be used to compute the average emission delay for one step. It is assumed the delay comprises the back-off delay (T_{back}) and the time required to appropriately receive the packet data packet from the receiver (T_{data}). It is the likelihood of an emergency message. For the sake of simplicity, assume that it is uniform for autos. τ Otherwise, and under typical conditions, the τ value is relatively tiny. n is the number of cars competing for access to the communication channel, including all neighbors within the range of telecommunication coverage R_{tr}).

According to (7), if a vehicle plans to transmit an emergency message, the number of unscreened applicants for channel access is defined as Nc.

$$N_c = n\tau + 1 \tag{7}$$

Back-off is meant to avoid interference when autos issue emergency notifications. If the vehicle interferes with the message when it receives it, it will reject it (the expression "back-off"). In this circumstance, the probability of a collision will be defined as (8).

$$PR_{w/0} = 1 - \frac{2w_{min}(n\tau + 1)}{(w_{min} + 1)^2 + 2w_{min}(n\tau + 1)} \tag{8}$$

in (8), w_{min} denotes the least conceivable CW size, which is often believed to be 32. The average number of transmitted entries will be specified in (9) concerning the maximum transmitting value (l_{re}) (3-7).

$$E(N) = \sum_{N=1}^{l_{re}} N\left(1 - Pr_{\frac{w}{0}}\right)^{N-1} Pr_{\frac{w}{0}} \tag{9}$$

If there is no interference channel, the CW will automatically double its maximum value $w_{max} = 2^m w_{win}$. Mistake a system parameter, which is presumed to be 5 in this case. As w_{con} approaches its maximum value of $w_{max}\ m + 1$, the CW will be used as the maximum in the following computations. As a result, (10) defines the average latency for a successful post.

$$T_{back} = \begin{cases} (2^{E(N)} - 1)w_{min}\,\eta\,; \ E(N) \le m \\ [(2^m - 1) + 2^m(E(N) - m)]w_{min}\eta; E(N) > m \end{cases} \tag{10}$$

η represents the length of the back-off slots. The length of the data transmission (T_{data}) is defined as the product of the size of the packet message packet (L_{size}) and the data transfer rate (M) and will be defined as (11).

$$T_{data} = \frac{L_{size}}{M} \tag{11}$$

Combining (10) and (11), we can define the delay in the transmission in the form of (12).

$$T_{trs} = T_{back} + \frac{L_{size}}{M} \tag{12}$$

In this part, a parameter called M8 will be added to define the relay node in the cluster to link the nodes inside the cluster to each other. The capabilities of neighbor and node relays will be fully utilized. The maximum delay of T_{max} is 100 ms based on the reference number. Also, to compute the delay factor in one step, the parameter D^* is introduced, which has a symmetrical connection with the T_{tr} emission delay parameter, so if $T_{tr}$is 0, $D^*$is equal to 1, and vice versa. In other words, the minor D^*, the smaller the larger the delayed T_{tr}release. As a result, in general, the parameter D^* will be specified by (13).

$$D^* = \begin{cases} 0\,; T_{tr} > T_{max} \\ 1 - \dfrac{T_{tr}}{T_{max}}\ ; else \end{cases} \tag{13}$$

The parameter D^* will have a value between 0 and 1. Furthermore, suppose the T tr diffusion delay in one step is more significant than T_{max}. In that case, the value of the parameter D^*will be 0, considered the worst-case scenario. Given the specified parameters and (12), the parameter valueM^* is defined as (14).

$$M^* = \alpha PR_{succ}(d) + \beta D^*\ ;\ \alpha, \beta \in [0,1], \alpha + \beta = 1 \tag{14}$$

in (14), α and β indicate the weighting factor of the propagation's reliability and latency properties. The parameter M^*'s of changes is similarly between 0 and 1. The more significant M^*symbolizes the sender's and receiver's safer and more acceptable communication. In the case of an incident and the necessity to transmit an emergency message in indirect mode, a node named "Node Relay" must be defined as the interface between the incident's ninety and the other nodes. In summary, the goal is to send an emergency message to V_i (as ninety) and other devices in the danger zone. In this scenario, the purpose is to identify a relay node (V_{relay}) among the automobiles inside the hazard range closest to the node (or, in other words, the lowest distance to the ninety of the incident). Furthermore, the M^* parameter between V_{tr} and V_{relay} should be bigger than M^*_{th}. (15) gives the characteristics for the clustering section with the rule R=XY.

$$P(R) = \frac{P(XY)}{|D|} \tag{15}$$

And the probability for the clustering section using weighted association rules is expressed as (16).

$$P_{WAssociationRules}(R) = \frac{P(XY)}{P(X)} \tag{16}$$

In the preceding calculations, P(.) is the number of data points from the complete set D that includes both X and Y. D in our suggested strategy is equivalent to the cluster from which the nearest data are chosen. The DSRC standard is split into seven 10 MHz frequency bands. The frequency band 178 manages other frequency bands to provide safe connections to the smart City-based Internet of Things.

4. Simulation and Results

Moving an automobile along a path in a smart city with the Internet of Things setup and infrastructure needs sensitive GPS information with the lowest possible latency and an improved signal-to-noise ratio. The problem of capturing vehicle distances and speeds on straight and steep mountain bumps must be solved to ensure that the network functions properly. Determining traffic bottlenecks, traffic, and any accidents over high terrain is difficult since it looks like a series of amplifiers are necessary. As a result, maintaining speed restrictions for autos in all scenarios is seen as critical. Another big issue, on the other hand, is the entry of automobiles into the tunnel. Satellite and GPS will not be able to transmit in such scenarios. Delays in data transfer and transmission will drop to a high level, as will the signal-to-noise ratio. Very little information will be available to the vehicles, and they may get zero kilobytes. This is particularly concerning in tunnels since the automobile receives no information. As a result, the suggested technique requires an amplifier every 20 meters within the tunnel, which may result in increased expenses for the Smart City with the Internet of Things setup and infrastructure, but can overcome this significant difficulty to some extent. As a result, the employment of the RTS and CTS concepts, as previously indicated, was examined in conjunction with channel estimates and the identification of communication channels. The data are examined, and graphs are generated to evaluate the suggested technique. The simulator utilized is NS-3, which excels at simulating computer networks. Installing the SUMO plugin's final version may also be called a Smart City with the Internet of Things setup and infrastructure, as shown in Fig. 9, located in Porto, Portugal.

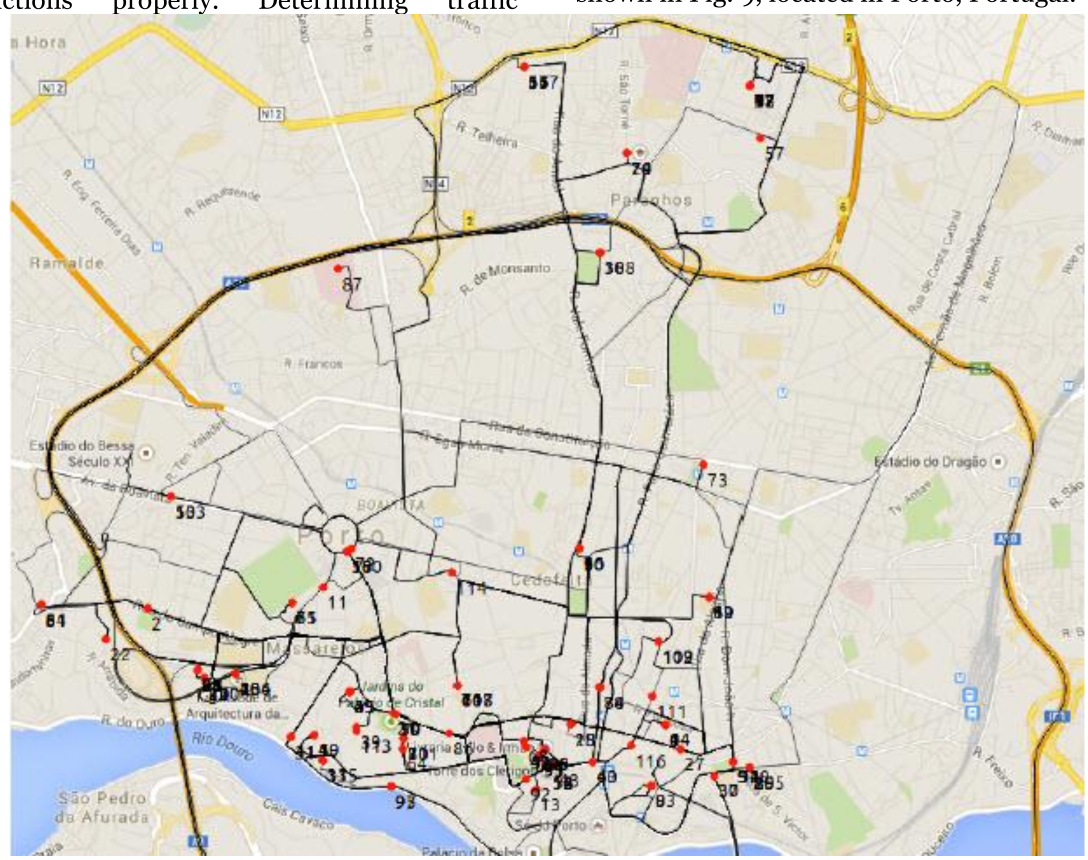

Fig. 9. View of the mountainous area of Porto in Portugal

It should be emphasized that not all of these pathways are necessary. However, sections of the mountains and the tunnel are also taken into account in this study. This route and map have been used for all assessments and outcomes. Similarly, Xerces C++ is regarded as a function to obtain maps that are more recent than Google, and the above map of Porto, Portugal, has been acquired in the same manner. GeographicLib, a short C++ class that has been used for geographical translation, UTM, UPS, MGRS, geocentric cartographic coordinates for gravity, geoid size, and geomagnetic field computations, is another library that has been utilized. Another package named libcurl is a C++ library for sending URLs from the client side. This library function can support the HTTPS protocol and read Google Maps APIs. The coordinates are taken into account when the map is opened. In Fig. 10, the coordinates in the target region may be determined by considering the X and Y coordinates as follows:

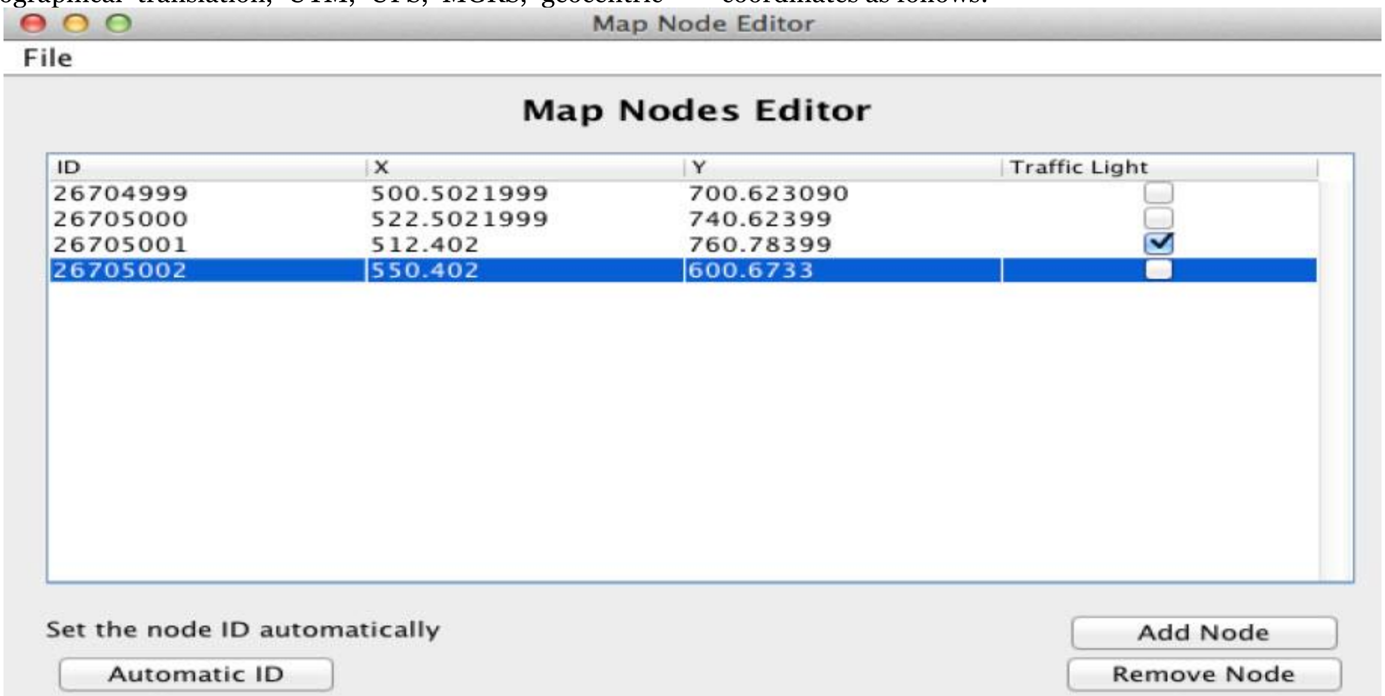

Fig. 10. X and Y map coordination

It will be feasible to compute the path precisely by introducing a Python script called randomtrips.py. Requests are a vital component for manually inserting cars on the route in the tunnel situation. It should be recognized using the coordinates entered in Fig. 10, as specified in the manner displayed in Fig. 11.

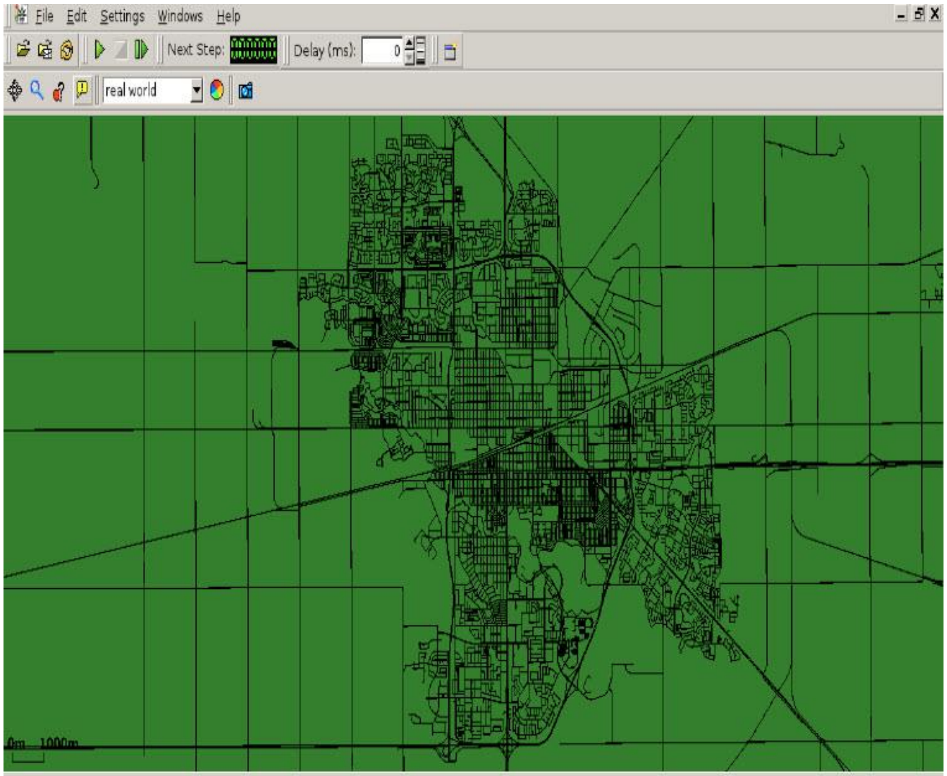

Fig. 11. I geographical coordinates are used to identify the area (a mountainous region located in a suburban city, in which the tunnel and the residential area are also scattered).

A specific mountain region must be defined to be presented in re-coordinates. This area will be increased, as illustrated in Fig. 12.

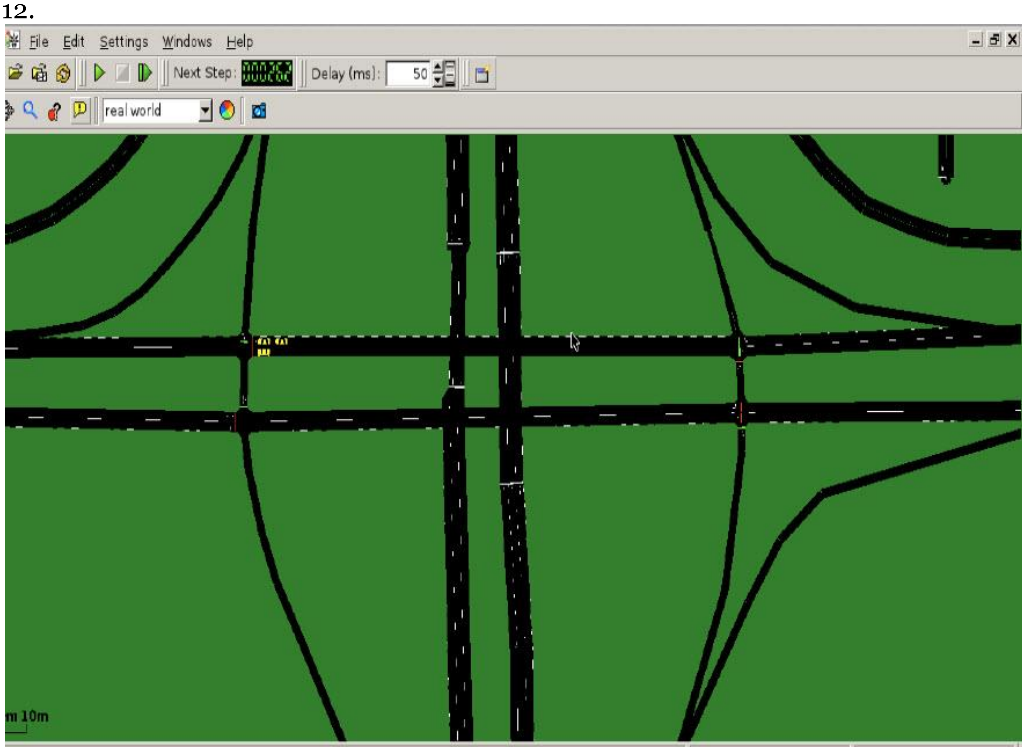

Fig. 12. Mountainous area and its roads

The road has two four-way lengths, as indicated in Fig. 12. In the case of a huge white dot, the components in the direct line are vertical, and there are tunnels. As indicated in Fig. 13, there should be a specific circumstance in the NS-Nam scenario.

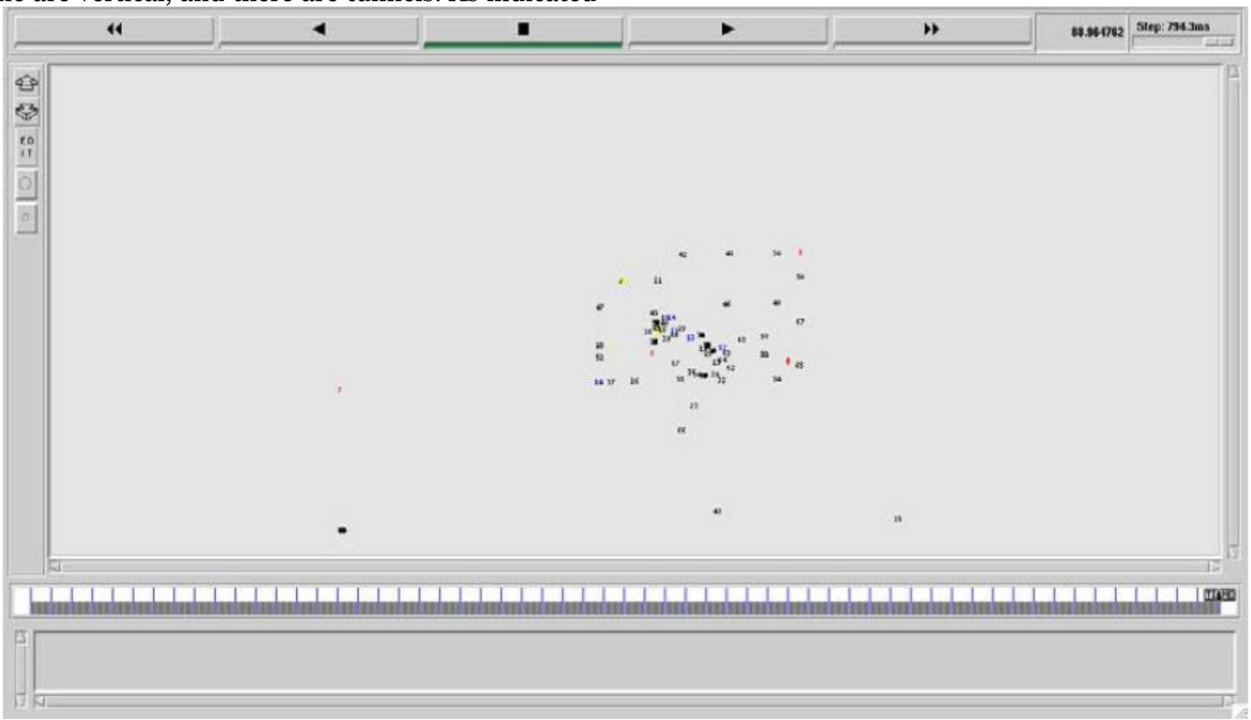

Fig. 13. Simulated scenario in NS-Nam

Specifying the Smart City's first parameters with the Internet of Things setup and infrastructure is required. Table 3 depicts the starting parameters of a smart city with the Internet of Things setup and infrastructure, as specified by a reference paper in this field.

Table 3. Smart City with Internet of Things configuration and infrastructure parameters

Network Dimension	$500x500\ m^2\ or\ \ 1000x1000\ m^2$
Each packet's size	1000 byte
Numbers of Vehicles	300
Cars with the slowest speeds (in mountainous roads)	30 km/h
Cars with the slowest speeds (in mountainous tunnels)	25 km/h
Each vehicle's radio radius	20m
Each vehicle's sender's energy	0.02 Jul
Each vehicle's receiver energy	0.04 Jul
Per-second fuel usage	2 sec
The number of available ports in a particular network setup	4000
The length of each path (on one side of the road)	5m
Modulation Schematic	OFDM
Data transmission power	21 dB
Rate of data transition	6 MB/s

In the OFDM channels, the modulation type is separated into two sections: QPSK on the road and 64QAM in the tunnels. At first, the packet rate, or PDR metric, will be shown as shown in Fig. 14.

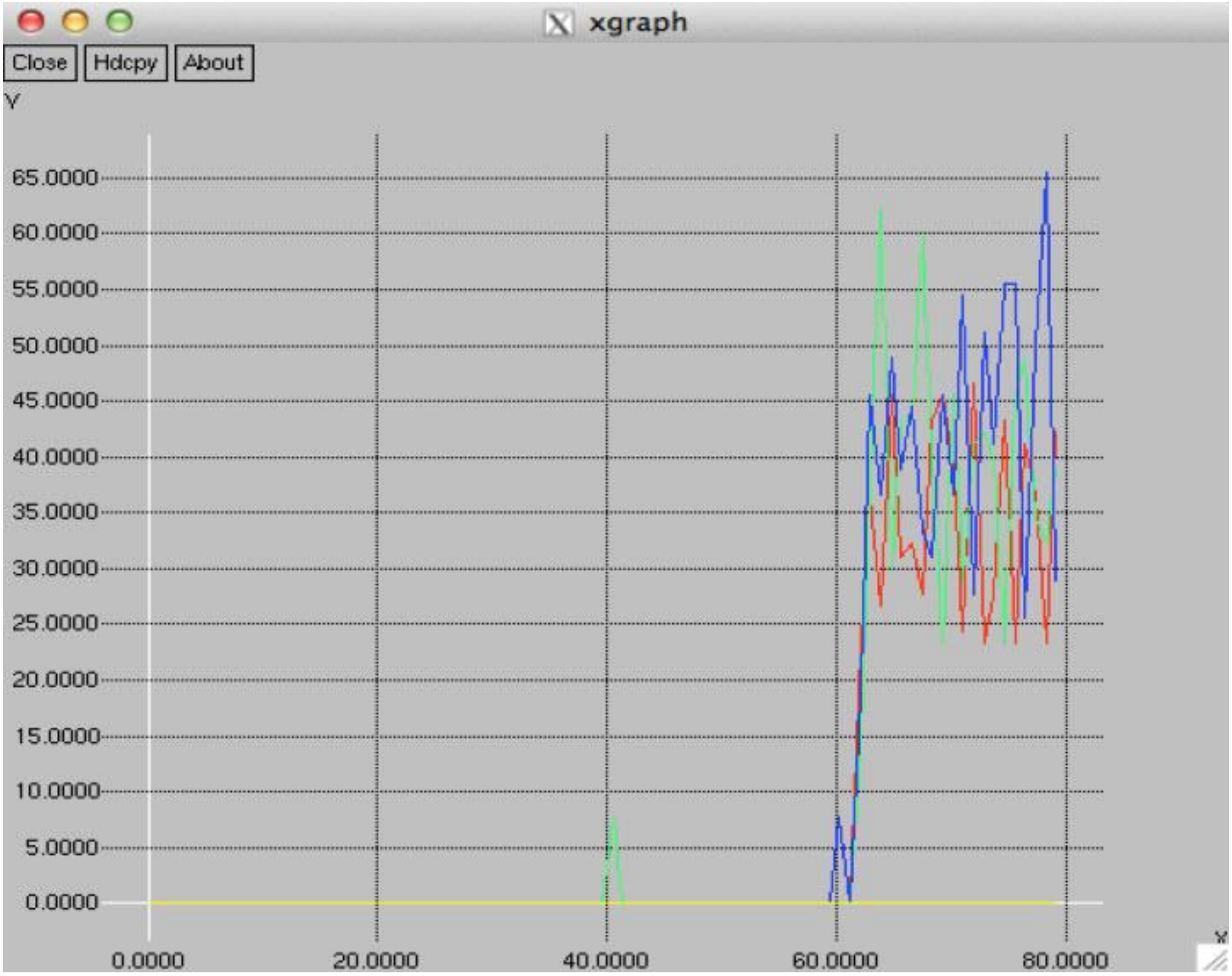

Fig. 14. PDR analysis

PDR is calculated by (17).

$$PacketDrop_{Calculation} = \frac{SentPacket\_Number}{RecievedPacket_{Number}} \quad (17)$$

The summary of (17) is also (18).

$$PacketDrop_{Calculation} = \left(\sum_{i=1}^{4} \frac{PacketDrop_{Number}}{SentPacket_{Number}}\right) \times 100 \quad (18)$$

i is the index of summation according to the above two relations. Fig. 6 depicts the packet loss rate under various traffic situations and tunnel congestion, which causes delays in transmitting and receiving data. When network resources are shared, the outcome reveals a high rate of PDR. PDR is evaluated at zero intervals; just one vehicle is in the covered zone and uses network resources (simulation start). Of course, this does not imply that the PDR will be assessed just for an automobile, even if a car fills the area. The end-to-end latency may be calculated using PDR and displayed as an analytical chart, as illustrated in Fig. 15.

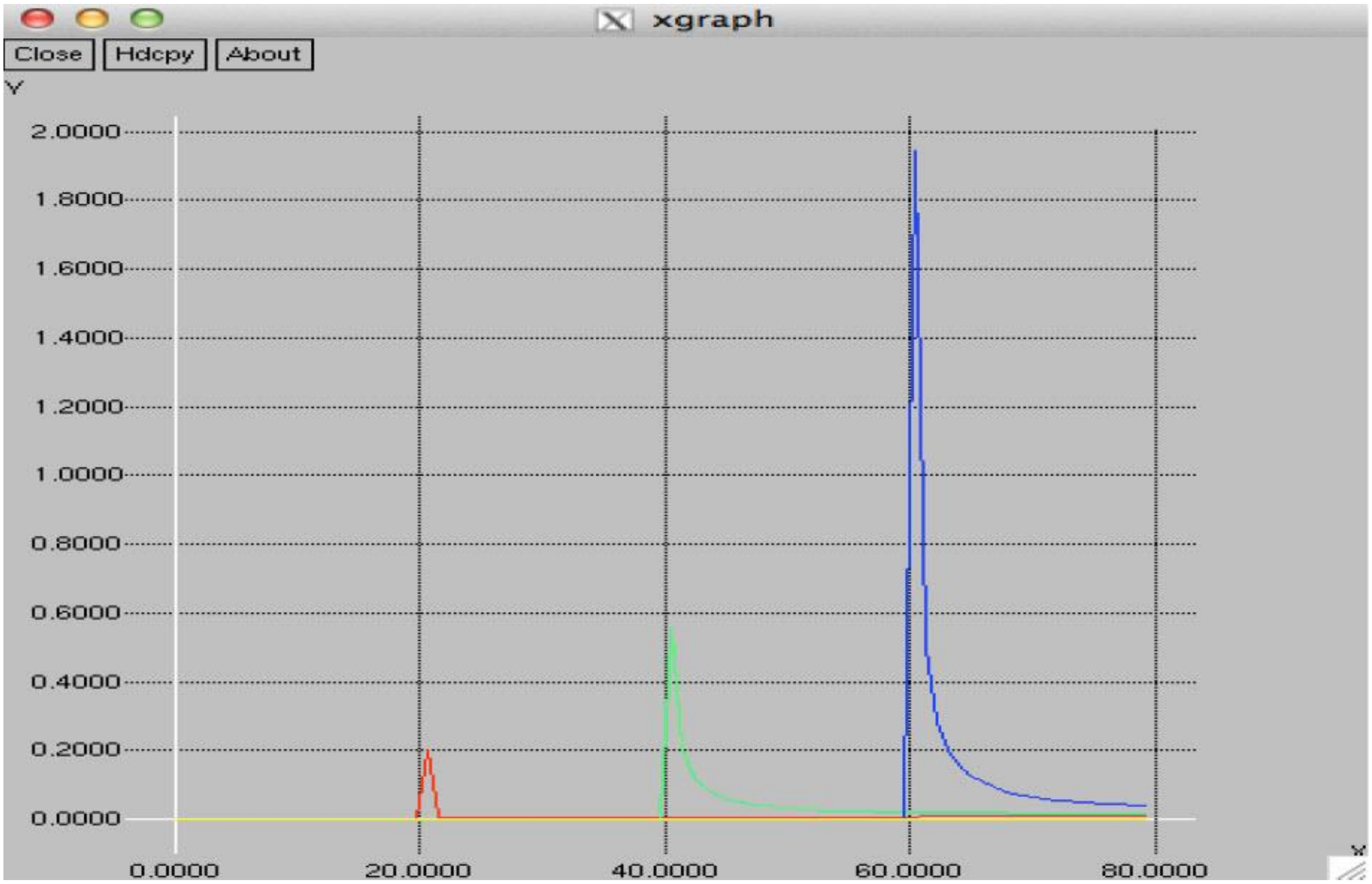

Fig. 15. End-to-end analysis

End-to-end latency analysis is obtained by keeping varied traffic circumstances in mind and is closely connected to PDR. The amount of congestion determines the end-to-end latency during routing and the time lag on the output channel for data transmission. (19) calculates the average end-to-end latency.

$$n - to - n_{delay} = \frac{decrease\ in\ delay}{4} \quad , \quad unit = ms \tag{19}$$

The simulation's throughput is determined depending on runtime, input condition, and vehicle unavailability in the network coverage region. Average throughput was computed using (20).

$$Throughput_{Avg} = \left(\sum_{i=1}^{4} \frac{RecievedPacket_{Number} \times PacketSize}{Total\ Simulation\ Time}\right) \times 100 \tag{20}$$

The output of throughput is like Fig. 16.

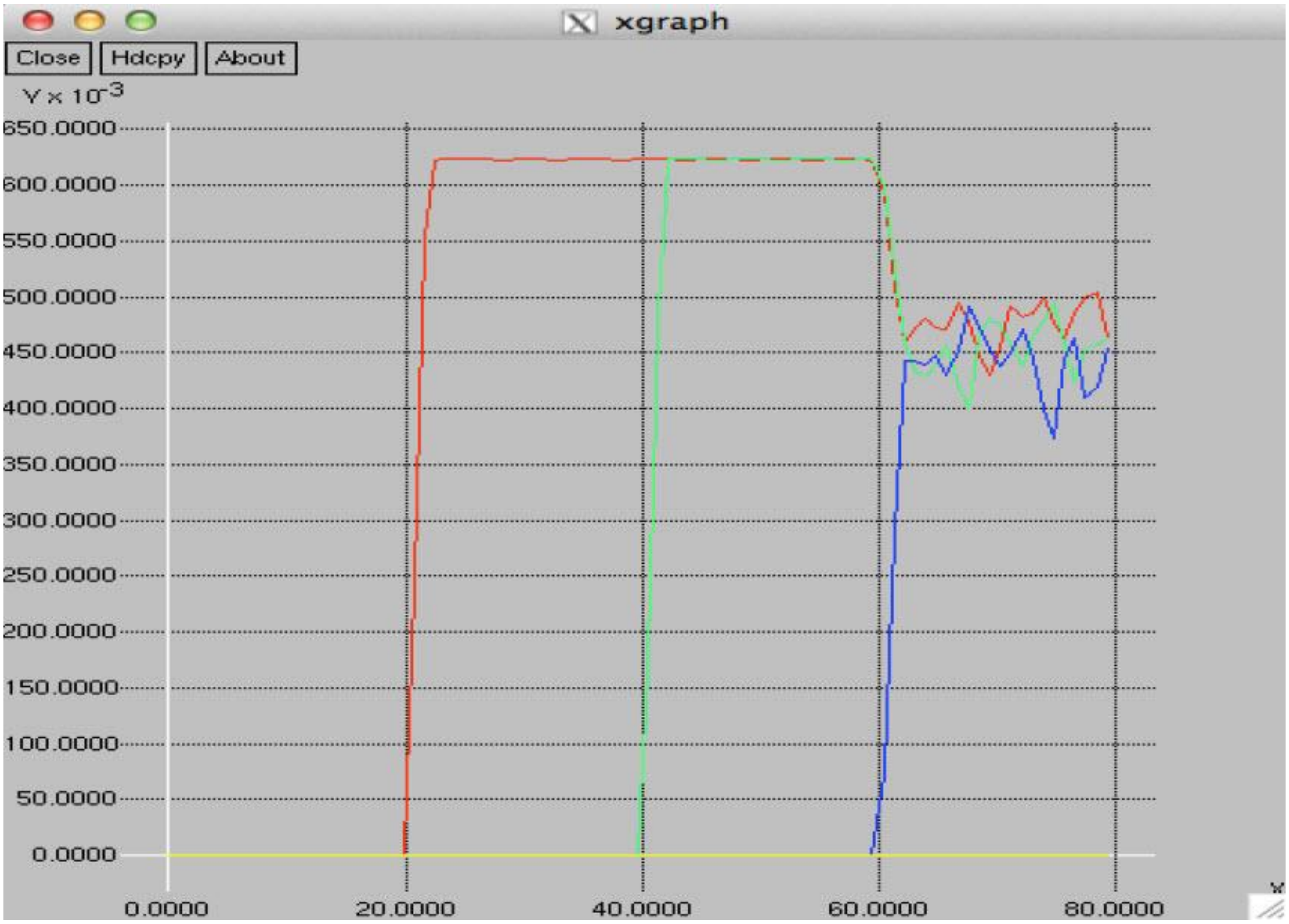

Fig. 16. Throughput analysis

According to an analysis, the ID 1 vehicle was withdrawn from the envelope in 1.1 seconds, whereas the vehicle with ID 2 began sending data in 20 seconds. The results of two-time delays demonstrate a significant decrease in bit rate. The NS section will review the work, which evaluates the evaluation by gathering this data and entering it into Excel software. Fig. 17 depicts a 300-second performance evaluation of an automotive network as the number of nodes or autos along with mountain route increases. The outcomes are as follows:

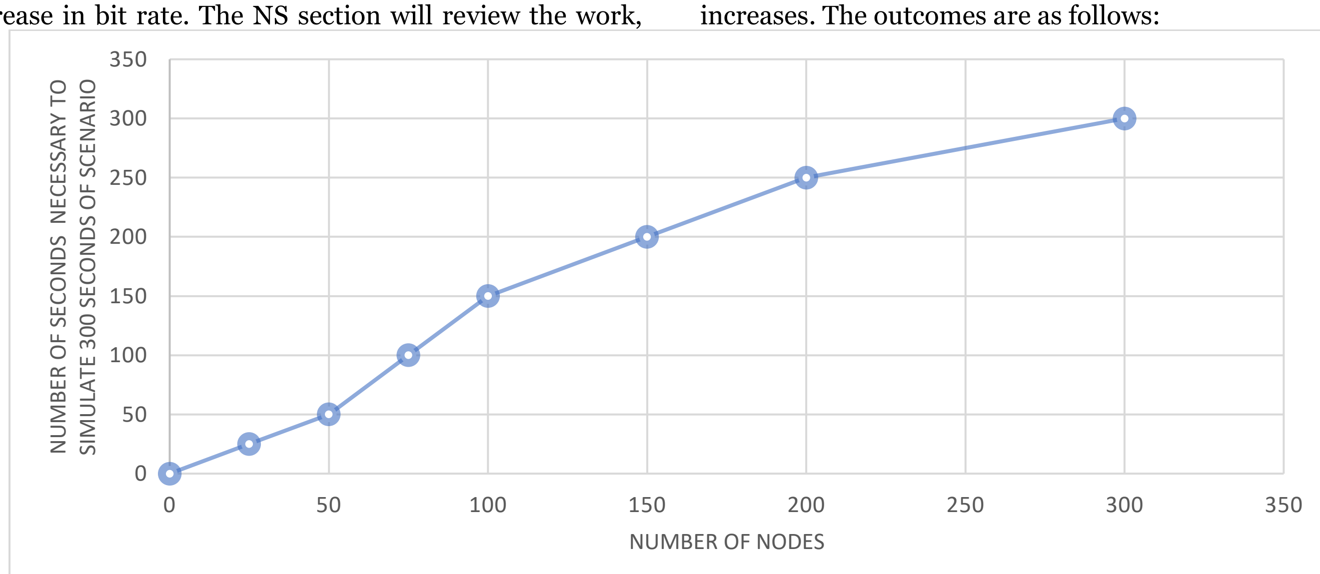

Fig. 17. Displays a performance evaluation of an automotive network in a 300-second scenario

As stated before, DSRC vehicles are the network's default protocol. The delivery and data loss rates of the protocol were compared to those of two other protocols, AODV and AOMDV, before a suggested method that utilized a weighted algorithm, SPA-(S, P), for improved resilience in any segment, was implemented. Results are shown in Figs. 18 and 19.

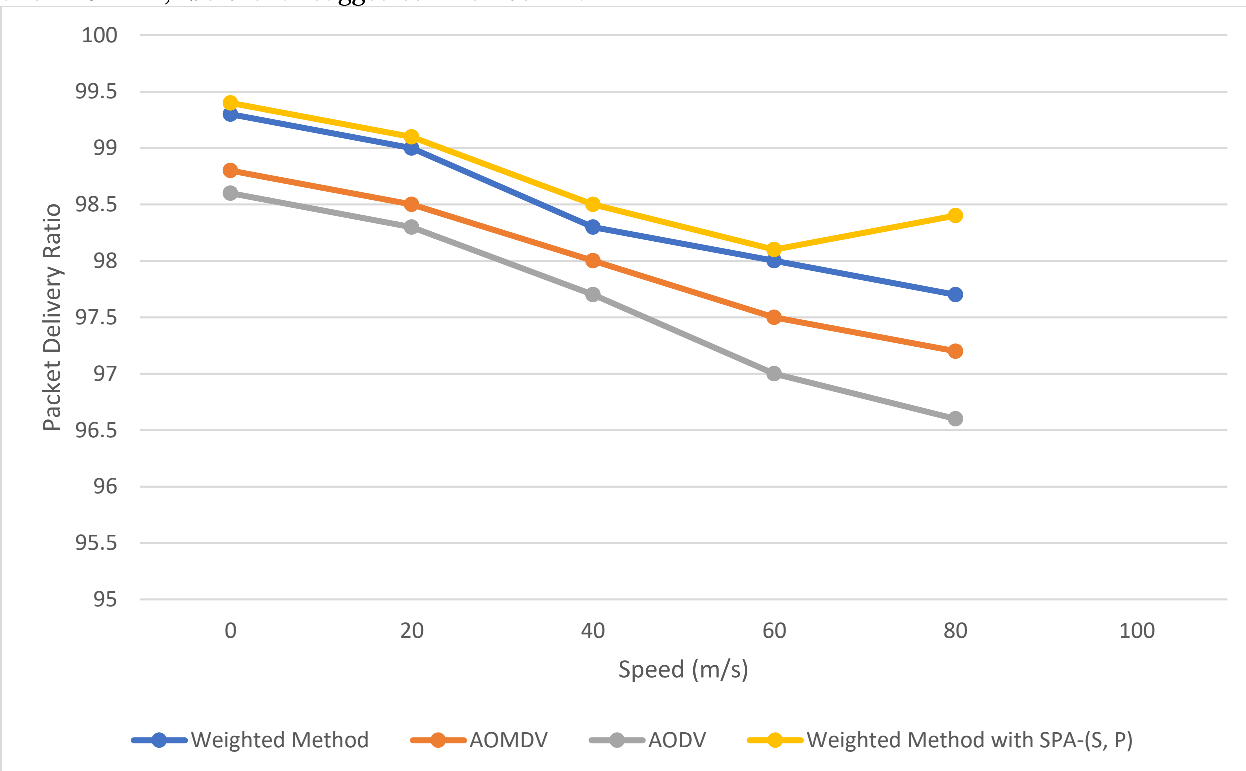

Fig. 18. Comparison of delivery rates of packages

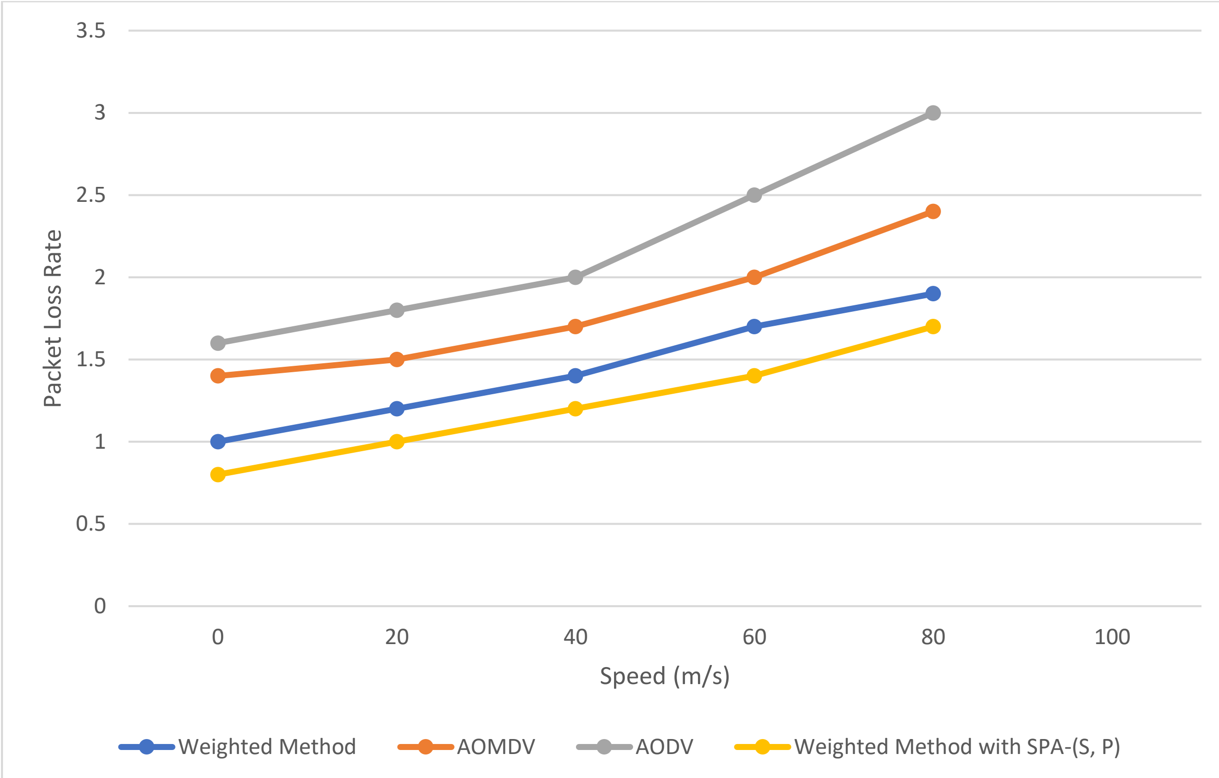

Fig. 19. Data loss rate or packet loss rate comparison

Fig. 18 shows that the proposed method outperforms the two AODV and AOMDV protocols in terms of packet delivery rate and is better adapted to deal with packet loss rates. As a consequence, data transmission and reception are quicker, and the permeability is higher. Fig. 20 displays a sustained routing comparison. The recommended routing stability protocol has been shown to be superior to the AODV and AOMDV protocols based on this outcome.

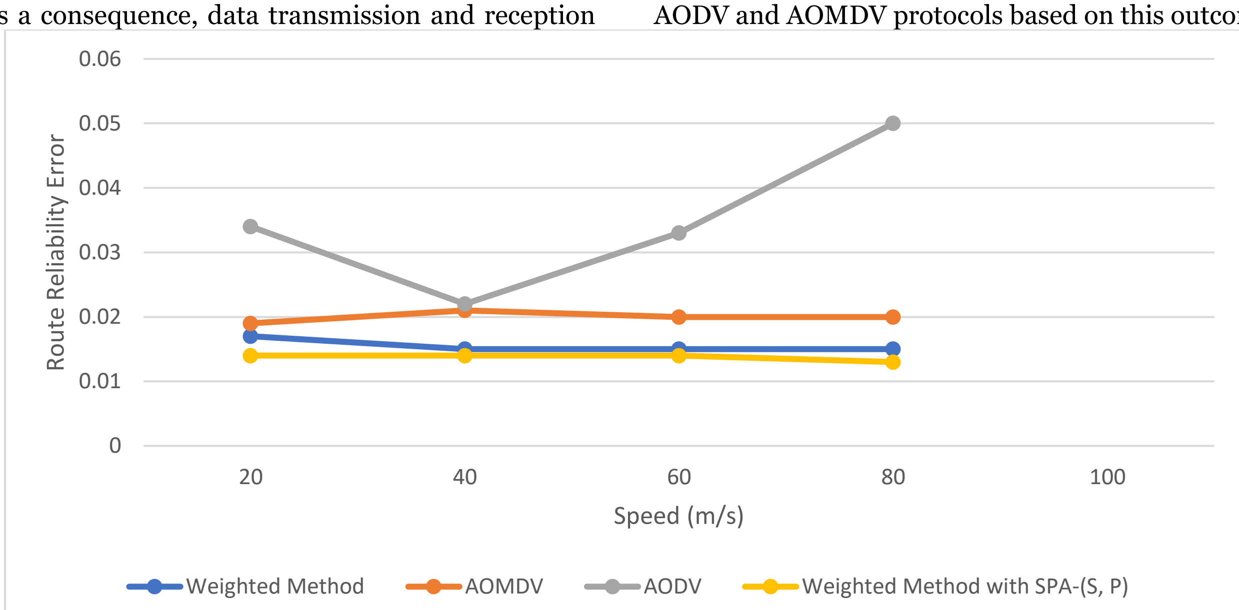

Fig. 20. A comparison of routing stability.

Because the modulation sections' output cannot be observed in the NS-3 environment, the current codes are brought into the MATLAB environment, and the results are also inserted in this part. This section has been added to the OFDM channels, and the modulation type is separated into two sections: QPSK on the road and 64QAM in the tunnels. Fig. 21 shows the bit error rate.

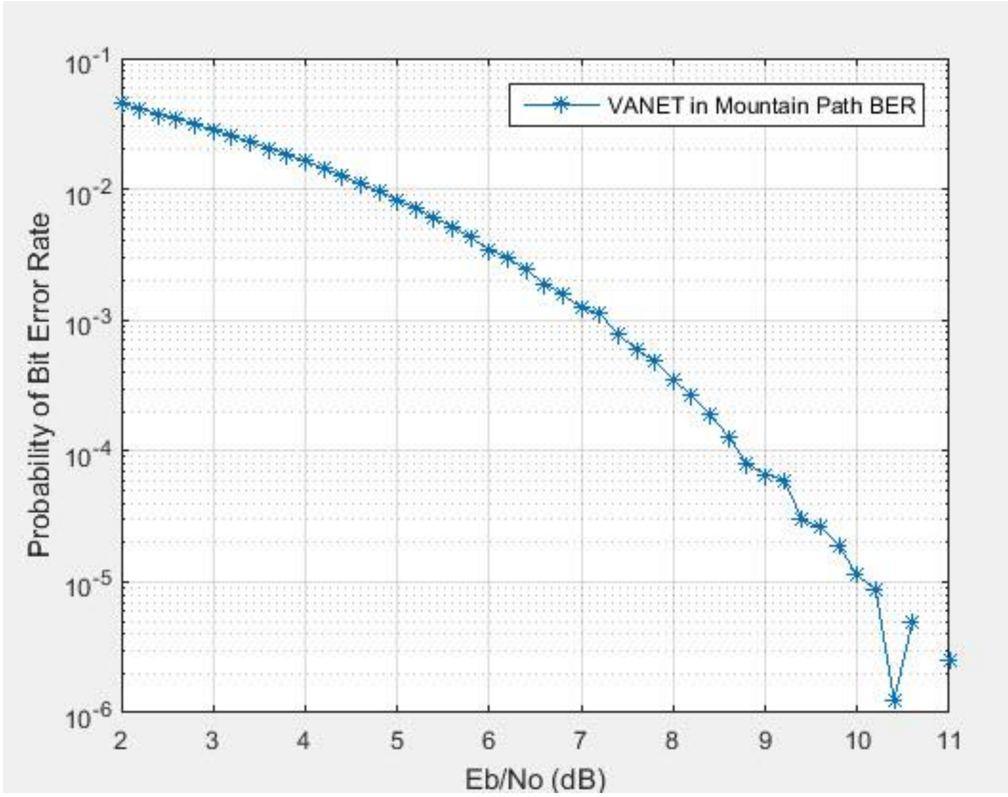

Fig. 21. Proposed method bit error rate

It looks like the bit error rate is decreasing. The graph shows that there is a low likelihood of bit errors on hilly paths by E b/N o and dB. Low does not always mean bad; rather, it denotes a reduction in data inaccuracy, which may include disturbances like noise. However, as can be seen at the bottom of this image, entering the tunnel caused a minor increase in bit error rate relative to the falling mountain's default state. This rise should not be disregarded. As seen in Fig. 22, when the bit error rate is decreased, the latency is also decreased.

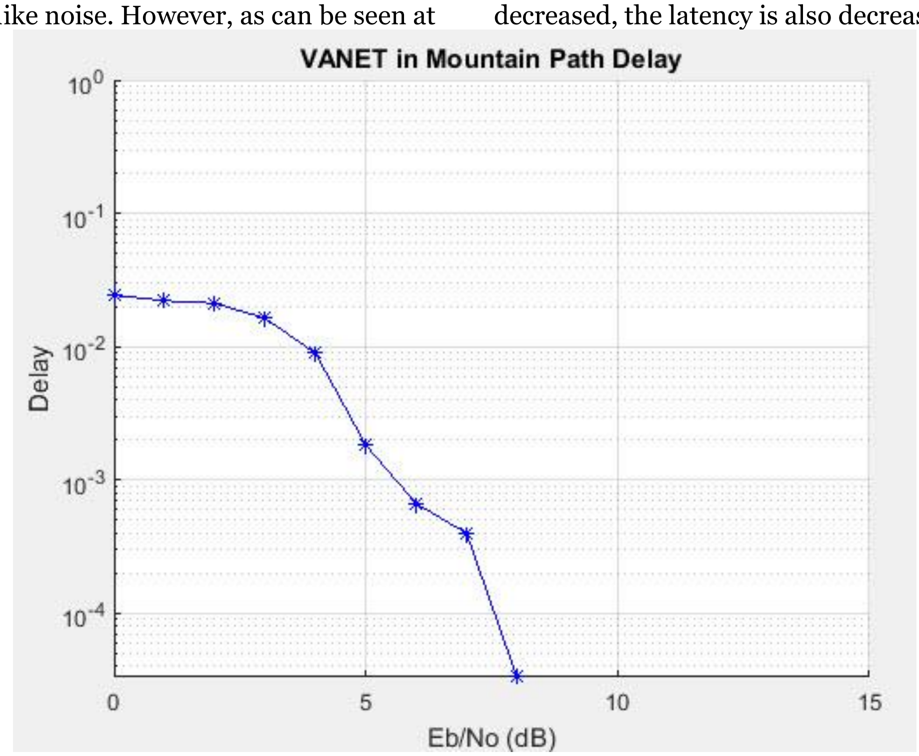

Fig. 22. Delay or latency rate

The delay is expressed in E_b/N_o units, indicating that it is minimized. Reduced bit error rates and delays will result in shorter latencies. Figs. 23 and 24 show this in the form of analysis (24).

Command Window

New to MATLAB? See resources for Getting Started.

```
Throughput     BER            Delay          Energy
2.000000     4.497875e-02   1.000000e+00     100
Throughput     BER            Delay          Energy
2.200000     4.135000e-02   1.000000e+00     100
Throughput     BER            Delay          Energy
2.400000     3.737750e-02   1.000000e+00     100
Throughput     BER            Delay          Energy
2.600000     3.471500e-02   1.000000e+00     100
Throughput     BER            Delay          Energy
2.800000     3.129500e-02   1.000000e+00     100
Throughput     BER            Delay          Energy
3.000000     2.840000e-02   1.000000e+00     100
Throughput     BER            Delay          Energy
3.200000     2.565500e-02   1.000000e+00     100
Throughput     BER            Delay          Energy
3.400000     2.302500e-02   1.000000e+00     100
```

Fig. 23. The statistical study results are throughput, bit error rate, latency, and energy in the simulated beginning time.

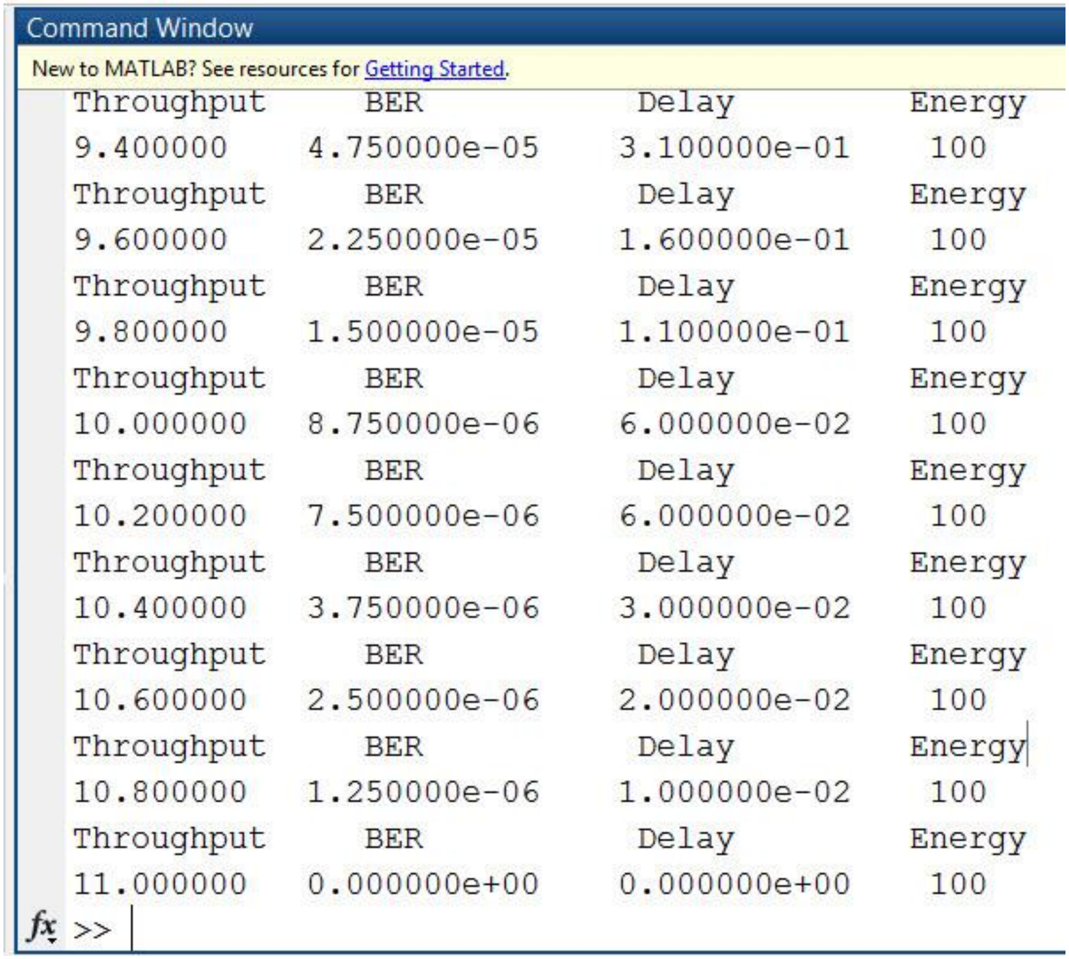

Fig. 24. Statistical analysis results include throughput, bit error rate, latency, and energy in the final simulation.

The network's principal energy was a total of 200 Jules. It is noted that energy is utilized at 100 Jules from the beginning to the conclusion of the simulation. Similarly, the permeation was based on Fig. 15 at the start of Simulation 2, which reached 11 at the end of the simulation, demonstrating development. The bit error rate began at four and ended at two. As cars enter the tunnel, bit rate error rates increase in portions of the bit rate, as illustrated in Fig. 13, which is evident in statistical analysis. The delay was there from the start of Simulation 1 and eventually dropped to zero. Nonetheless, the ups and downs of high-altitude access and tunnels have implications.

5. Conclusion

In recent years, the number of vehicles in metropolitan areas has increased substantially. Accidents have grown as a result of increased traffic. Despite the introduction of different accident detection devices to the market, a significant number of fatalities occur. The problem stems from a failure to respond to significant accidents on time, which is caused by insufficient automated accident detection and inefficient emergency response communication and routing. The lack of efficient pricing and retrofit capacity systems exacerbates the problem significantly. We propose an IoT-based accident detection system to solve these problems. We demonstrated how using various sensors may help more precisely recognize a traffic incident. The proposed system immediately identifies an accident, locates the nearest hospital, and sends an emergency help request to the relevant hospital department. This technology makes a decision based on data from smartphone sensors that detect information about the vehicle's status. We proved that our suggested method minimizes the number of false alarms. Our system requires Internet connectivity to function correctly. One disadvantage of our study is that we conducted the system's primary assessment in a simulated circumstance. We want to improve the system by implementing mobile edge computing to minimize latency while boosting security and privacy. Indeed, the technique requires a thorough examination.

REFERENCES

[1] W. H. Organization, *Global status report on road safety 2015*. World Health Organization, 2015.

[2] Asirt.org, "Road Crash Statistics," *Available online: https://www.asirt.org/safe-travel/road safety-facts/, (accessed on 30 January 2019)..*

[3] A. Khan, F. Bibi, M. Dilshad, S. Ahmed, Z. Ullah, and H. Ali, "Accident detection and smart rescue system using Android smartphone with real-time location tracking," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 6, pp. 341–355, 2018.

[4] L. Jackson and R. Cracknell, "Road accident casualties in Britain and the world," *House of Commons Library: London, UK*, 2018.

[5] IEEE. Pbs.gov.pk, "Traffic Accidents (Annual) Pakistan Bureau of Statistics.," *Available online: http://www.pbs.gov.pk/content/traffic-accidents-annual (accessed on 30 January 2019)..*

[6] J. Cuena, J. Hernández, and M. Molina, "Knowledge oriented design of an application for real time traffic management: The TRYS system," in *European Conference on Artificial Intelligence (12th: 1996: Budapest, Hungary). ECAI 96*, 1996.

[7] S. Inoue, K. Shozaki, and Y. Kakuda, "An automobile control method for alleviation of traffic congestions using inter-vehicle ad hoc communication in lattice-like roads," in *2007 IEEE Globecom Workshops*, IEEE, 2007, pp. 1–6.

[8] F. Bhatti, M. A. Shah, C. Maple, and S. U. Islam, "A novel internet of things-enabled accident detection and reporting system for smart city environments," *sensors*, vol. 19, no. 9, p. 2071, 2019.

[9] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[10] S. Radomirovic, "Towards a model for security and privacy in the internet of things," in *Proc. First Int'l Workshop on Security of the Internet of Things*, 2010, p. 6.

[11] P. P. Ray, "A survey on Internet of Things architectures," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 3, pp. 291–319, 2018.

[12] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," *Journal of network and computer applications*, vol. 98, pp. 27–42, 2017.

[13] O. Salman, I. Elhajj, A. Chehab, and A. Kayssi, "IoT survey: An SDN and fog computing perspective," *Computer Networks*, vol. 143, pp. 221–246, 2018.

[14] P. Zhang, M. Zhou, and G. Fortino, "Security and trust issues in fog computing: A survey," *Future Generation Computer Systems*, vol. 88, pp. 16–27, 2018.

[15] V. K. Arora, V. Sharma, and M. Sachdeva, "On QoS evaluation for ZigBee incorporated Wireless Sensor Network (IEEE 802.15. 4) using mobile sensor nodes," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 2, pp. 27–35, 2022.

[16] F. K. Shaikh and S. Zeadally, "Energy harvesting in wireless sensor networks: A comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 1041–1054, 2016.

[17] A. More and V. Raisinghani, "A survey on energy efficient coverage protocols in wireless sensor networks," *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 4, pp. 428–448, 2017.

[18] S. He, S. Banerjee, and K. Sun, "Can derivative determine the dynamics of fractional-order chaotic system?," *Chaos Solitons Fractals*, vol. 115, pp. 14–22, 2018.

[19] L. Yuan and Q. Yang, "Parameter identification of fractional-order chaotic systems without or with noise: Reply to comments," *Commun Nonlinear Sci Numer Simul*, vol. 67, pp. 506–516, 2019.

[20] Y. Scharf, "A chaotic outlook on biological systems," *Chaos Solitons Fractals*, vol. 95, pp. 42–47, 2017.

[21] R. Saborido and E. Alba, "Software systems from smart city vendors," *Cities*, vol. 101, p. 102690, 2020.

[22] P. Chamoso, A. González-Briones, F. De La Prieta, G. K. Venyagamoorthy, and J. M. Corchado, "Smart city as a distributed platform: Toward a system for citizen-oriented management," *Comput Commun*, vol. 152, pp. 323–332, 2020.

[23] Y. Zhao, "Mobile phone location determination and its impact on intelligent transportation systems," *IEEE Transactions on intelligent transportation systems*, vol. 1, no. 1, pp. 55–64, 2000.

[24] M. Reddy and J. Tulasi, "Accident detection depending on the vehicle position and vehicle theft tracking, reporting systems," *Int. J. Sci. Eng. Technol. Res*, vol. 3, no. 9, pp. 2359–2362, 2014.

[25] H. M. Ali and Z. S. Alwan, *Car accident detection and notification system using smartphone*. Lap Lambert Academic Publishing Saarbrucken, 2017.

[26] K. Patel, "Utilizing the emergence of android smartphones for public welfare by providing advance accident detection and remedy by 108 ambulances," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, no. 9, pp. 1340–1342, 2013.

[27] F. Aloul, I. Zualkernan, R. Abu-Salma, H. Al-Ali, and M. Al-Merri, "iBump: Smartphone application to detect car accidents," *Computers & Electrical Engineering*, vol. 43, pp. 66–75, 2015.

[28] I. Khot, M. Jadhav, A. Desai, and V. Bangar, "Go Safe: Android application for accident detection and notification," *Int. Res. J. Eng. Technol*, vol. 5, no. 5, pp. 4118–4122, 2018.
[29] J. Zaldivar, C. T. Calafate, J. C. Cano, and P. Manzoni, "Providing accident detection in vehicular networks through OBD-II devices and Android-based smartphones," in *2011 IEEE 36th Conference on Local Computer Networks*, IEEE, 2011, pp. 813–819.
[30] A. Bin Faiz, A. Imteaj, and M. Chowdhury, "Smart vehicle accident detection and alarming system using a smartphone," in *2015 international conference on computer and information engineering (ICCIE)*, IEEE, 2015, pp. 66–69.
[31] C. Thompson, J. White, B. Dougherty, A. Albright, and D. C. Schmidt, "Using smartphones to detect car accidents and provide situational awareness to emergency responders," in *Mobile Wireless Middleware, Operating Systems, and Applications: Third International Conference, Mobilware 2010, Chicago, IL, USA, June 30-July 2, 2010. Revised Selected Papers 3*, Springer, 2010, pp. 29–42.
[32] A. Rajkiran and M. Anusha, "Intelligent automatic vehicle accident detection system using wireless communication," *Int. J. Res. Stud. Sci. Eng. Technol*, vol. 1, no. 8, pp. 98–101, 2014.
[33] N. H. Sane, D. S. Patil, S. D. Thakare, and A. V Rokade, "Real time vehicle accident detection and tracking using GPS and GSM," *Int. J. Recent Innov. Trends Comput. Commun*, vol. 4, no. 4, pp. 479–482, 2016.
[34] C. Prabha, R. Sunitha, and R. Anitha, "Automatic vehicle accident detection and messaging system using GSM and GPS modem," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 3, no. 7, pp. 10723–10727, 2014.
[35] N. Dogru and A. Subasi, "Traffic accident detection using random forest classifier," in *2018 15th learning and technology conference (L&T)*, IEEE, 2018, pp. 40–45.
[36] A. Fanca, A. Puscasiu, S. Folea, and H. Vălean, "Trauma accident detecting and reporting system," in *2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, IEEE, 2018, pp. 1–5.
[37] R. A. Young, "Association between embedded cellular phone calls and vehicle crashes involving airbag deployment," in *Driving Assessment Conference*, University of Iowa, 2001.
[38] Y.-K. Ki and D.-Y. Lee, "A traffic accident recording and reporting model at intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 188–194, 2007.
[39] D. B. Tushara and P. A. H. Vardhini, "Wireless vehicle alert and collision prevention system design using Atmel microcontroller," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, IEEE, 2016, pp. 2784–2787.
[40] M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni, "A system for automatic notification and severity estimation of automotive accidents," *IEEE Trans Mob Comput*, vol. 13, no. 5, pp. 948–963, 2013.
[41] N. Chaturvedi and P. Srivastava, "Automatic vehicle accident detection and messaging system using GSM and GPS modem," *Int. Res. J. Eng. Technol.(IRJET)*, vol. 5, no. 3, pp. 252–254, 2018.
[42] G. Liang, "Automatic traffic accident detection based on the internet of things and support vector machine," *Int. J. Smart Home*, vol. 9, no. 4, pp. 97–106, 2015.
[43] J. Maleki, E. Foroutan, and M. A. Rajabi, "Intelligent alarm system for road collision," *Journal of Earth Science and Engineering*, vol. 1, no. 3, 2011.
[44] E. Nasr, E. Kfoury, and D. Khoury, "An IoT approach to vehicle accident detection, reporting, and navigation," in *2016 IEEE international multidisciplinary conference on engineering technology (IMCET)*, IEEE, 2016, pp. 231–236.