Aditya Subramanian *et al*, International Journal of Computer Science and Mobile Applications,
Vol.6 Issue. 4, April- 2018, pg. 10-18 **ISSN: 2321-8363**
Impact Factor: 5.515

Intelligent Road Accident Detection and Response using IoT (Internet of Things)

Aditya Subramanian[1], Sanjeev Rao[2], Sourav Patra[3], Mrs. Visnudharsini S[4]
[1]Student, Department of CSE, SRM University, Ramapuram, aditya21subu@gmail.com
[2]Student, Department of CSE, SRM University, Ramapuram, rao.sanjeev0@gmail.com
[3]Student, Department of CSE, SRM University, Ramapuram, patrasouravskp@gmail.com
[4]Professor, Department of CSE, SRM University, Ramapuram, visnu.sudar@gmail.com

Abstract

According to a 2013 WHO report, road accidents are a leading cause of death in India, with over 130,000 deaths in the year alone. We believe that a major reason is the dependence on the human element - namely, requiring bystanders to help report accidents. In effect, automating the reporting process and hence effectively removing the need for witnesses to the scene will help improve outcomes. In this paper, we propose a next-generation IoT-enabled system to automate the reporting process, which should help reduce fatalities due to road accidents.

Keywords*:* IoT, WoT, sensors, accident, prevention, road accident, accident detection.

1. Introduction

Road accidents are a leading cause of death in urban areas in India. In recent years, there has been a rise in the number of road accidents, resulting in the loss of many lives due the lack of a timely detection and response system. According to the SaveLIFE foundation, in 2016, there was a total of about 150,000 deaths in the whole of India due to road accidents [1]; the major reason for such high figures is the high response time between the accident occurring and the respective authorities arriving, which has led to the loss of many lives. The leading cause of accidents are drunken driving, running red lights and driving on the wrong side, amongst others. A delay is usually caused by communication gaps between witnesses and the hospitals, or the bystander effect, or others. So far, there are few systems to combat this problem which have been able to significantly reduce the response time. This paper proposes a system which aims to reduce the time delay involved, using a semi-automated system, incorporating CCTV cameras and pressure and airbag sensors to detect an accident. Upon detection of an accident, footage from the CCTV camera is sent to the control room, where the footage is reviewed, and a decision is made as to whether the police should be called – if it is a minor accident – or an ambulance will be informed to a nearby hospital as well as the police if it is a major accident. It is expected that such a system will help reduce the number of fatalities incurred.

Section 2 of this paper explains the existing system present. Section 3 is a brief literature survey of the systems present; Section 4 then explains the system in detail, with the modules and the overarching description and working. Section 5 describes how the implementation is made using the hardware. Finally, section 6 (conclusion & future work) concludes the paper.

2. Existing System

The existing system relies mostly on the help of witnesses or bystanders to the accident scene, who in turn contact the ambulance, police and/or other relevant authorities regarding the accident. This method is prone to a few flaws, insofar that the responsibility is shouldered mostly by the bystander: witnesses must wait for the ambulance or police to arrive and will possibly have to give a statement. They may be further questioned about the accident, which is a process that can take up their time (which they may have planned for something else), and so they may avoid taking part due to the hassle involved, especially since most citizens do not know about the Good Samaritan act (which aims to protect the rights of those helping in an accident, but is not widely known about.)

Another potential pitfall is that of the bystander effect, wherein inaction occurs amongst a crowd, primarily due to reduced individual responsibility - where each assumes the other takes charge of the situation. Anecdotally, there have also been cases where the victims were attacked by the bystanders.
It is evident, then, that it is imperative to reduce involvement of third parties as much as possible: that way, detection is still possible in remote or hostile locations where not many are able to help the victims of the accident.

3. Related Works & Literature Survey

In this section the related works that prompted this paper will be examined briefly. The prior publication by Nasr et al., titled "An IoT Approach to Vehicle Accident Detection, Reporting, and Navigation" [2] examines a similar premise: namely, that of the usage of IoT (Internet of Things)-enabled devices to facilitate timely responses to accidents. However, it does not cater to all users, as it depends on user having a smartphone which also supports near-field communication (NFC). This is purportedly done to help identify the user(s) involved in an accident; however, while the market penetration of smartphones in India is quite high, the number of phones with NFC capabilities lower and reserved to higher end phones. Furthermore, this information is believed be extraneous because the identification of the users involved can be performed; a third party aggregating people's biometric data (such as blood type, etc.) lends itself to various privacy related concerns. As a result, it is believed that by removing the dependency on users to manually activate the system by making use of their NFC phone, a smoother system can be implemented which is essentially transparent to the user, once installed.

Another paper, "Novel drunken driving detection models using Internet of Things", [3] was analysed. In this paper, the accident prevention is done on the assumption that drunken driving and distracted driving are the main reasons for accidents, which does not cover the other accident reasons. Though the idea behind this system is sound, it has a couple of flaws in that it also does not prevent the user from driving the car once it has already started, e.g. if another person has started the car on their behalf. The system can further be enhanced by making it a multistage system for the prevention of accidents.

In a similar vein, "Mission On! Innovations in Bike Systems to Provide a Safe Ride based on IoT", [4] this paper considers the accidents for the prevention of accident for bikers. The bike in this system can only be started when the biker has worn their helmet, but, like the previous system, it does not detect accidents; it merely prevents the damage done because of an accident (since helmets significantly reduce fatalities for bikers). Like the previous system, this system can further have improved by adding other modules for different type of vehicles and by incorporating it into another system, as a part of a whole rather than a standalone system.

In "Intelligent safety Information Gathering System Using a Smart Black box" [5] this paper uses a device called a black box which is then extended to gather information about license plates of cars, which may be forensically useful, but does not detect or prevent an accident otherwise. The system can further be improved by making the information system a part of the proposed system and using the details in this paper to better automate this system.

4. Method & System Design

This section will describe the process of the system as a function of their individual components. It consists of three modules, each of which are involved in a separate "phase"; the overall architecture diagram is shown below.

Aditya Subramanian *et al*, International Journal of Computer Science and Mobile Applications,
Vol.6 Issue. 4, April- 2018, pg. 10-18 **ISSN: 2321-8363**
Impact Factor: 5.515

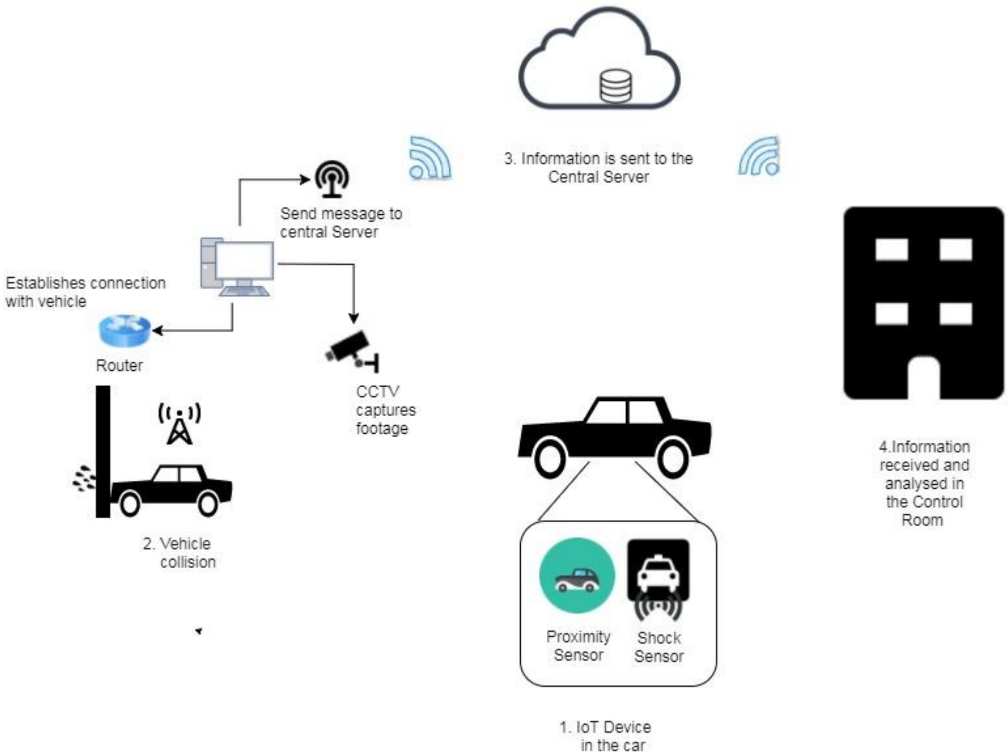

Figure 1: Architecture diagram of system

The phases are:

Backend registration: The back-end registration module involves the registration of the control rooms setup near the parts of the city monitoring the accidents (which will be henceforth referred to as monitoring-enabled locations, or MEL for short). It is not required for a control room to be present near every monitoring-enabled location, since the closest control room to an MEL will be selected whenever an incident occurs, I.e. the relationship between a control room and an MEL is one-to-many.

The reason a control room has to be registered is because the MEL has to query the central server for the list of all known control rooms near it; since the MEL is responsible for broadcasting an accident event, the control room has no (and does not need to have any) knowledge of the various MELs scattered throughout the city; without registering the control room, it will not be visible to the MEL and hence will detect accidents, but will not make their presence known.

The registration involves inputting the various details of the control room (such as manned capacity - the max number of accident events that can be handled by the control room and address and GPS location of the control room, used by the MEL to decide the closest control room; additional data can be added as necessary) which is then stored on an XML file (available in JSON format for redundancy) on the central server.

User registration: While this section is not strictly a separate module, it serves an important role in the detection of an accident. The detection of the accident is enabled by the user of a shock-proximity sensor duo, which must be installed on the car prior to the accident, controlled by an Arduino Uno. These sensors, along with the Arduino Uno, will be featured in the next module, the accident detection module.

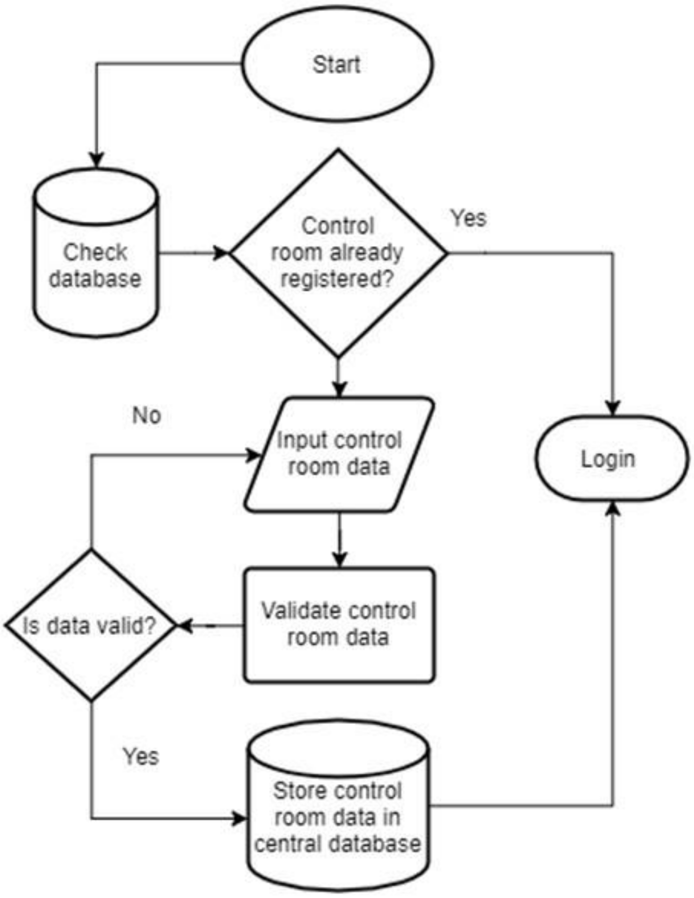

Figure 2: Registration module

The MEL consists of a series of wireless routers placed along fixed intervals alongside the stretch of the road where monitoring is to take place, and computers connected to these routers. Ideally, the routers should be placed such that there is as little overlap between routers, so that there is a one-to-one correspondence between routers and computers. The detection system installed (during user registration) on the vehicle automatically connects to the routers; these are, in turn, connected to the central server via the internet. The computers detect messages coming from the detection devices and send messages to the central server, effectively acting as a mediator.

The shock-proximity sensor pair is basically as the name suggests, a shock sensor (or detector) and a proximity sensor installed on the front of the car. The shock detector is triggered whenever any objects (usually other vehicles, static objects like walls or trees, or even dynamic objects like pedestrians) collide with the bumper of the car. The proximity sensor is triggered whenever any object approaches its range; this serves as a backup in case the shock detector gets destroyed before it has a chance to report back to the system. The algorithm is described below:

1. Proximity sensor detects object approaching it.
2. If there is no collision reported by the shock sensor, but the shock sensor works (i.e. reports HIGH whenever no shock detected) then assume no accident has occurred and continue.
3. If there is a collision reported by the shock detector (reports LOW) then assume a possible accident (or incident) has occurred.
4. If the shock sensor is unreachable then assume an incident has occurred.

All this is reported by the Arduino Uno, to the MEL.

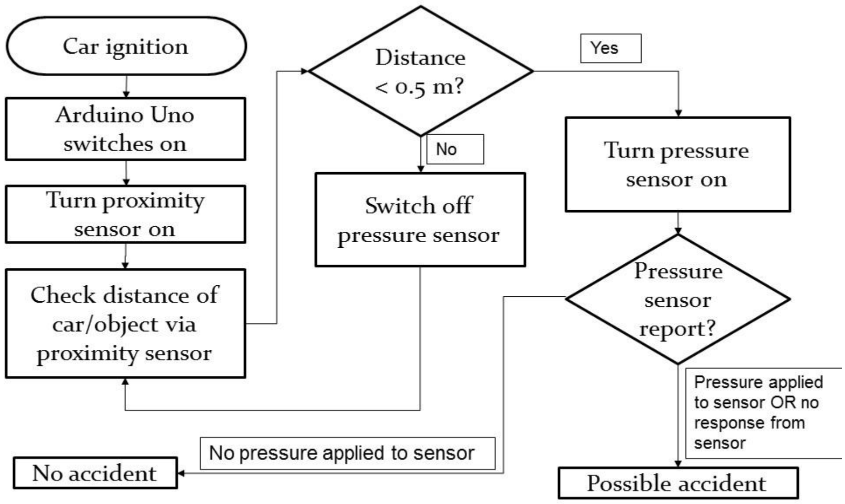

Figure 3: Detection module

Accident response: Once an accident has been detected, the controller sends a broadcast message to the router, which is then intercepted by a computer present in the MEL; this then sends a HTTP request to the central server, with the location data included in the request; it also saves the last 5 minutes' worth of video footage captured by the CCTV camera present to disk. The HTTP request is then processed by the central server, which locates the nearest control room to the MEL, and displays the video footage from the CCTV onto the control room's screens.

From there, the control room operators are alerted, at which point they can review the footage and decide whether an accident has occurred or not. If an accident has not occurred, they discard the footage; or else, based on the severity of the accident (involving casualties or otherwise) call the relevant authorities (police, hospitals, etc.); since the operators are alerted about an incident, it is not required for an operator to be present in the control room all the time.

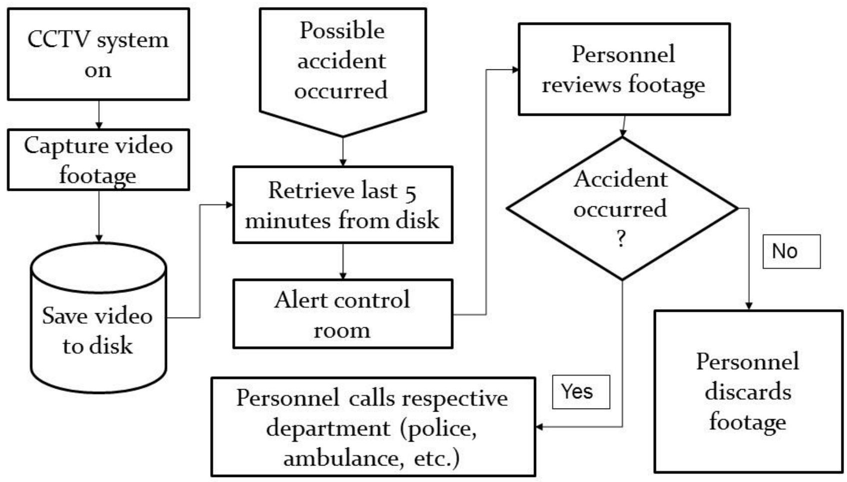

Figure 4: Response module

5. Implementation Details

In this section, the hardware and the software used to implement the system will be listed along with a brief reason for their usage.

5.1 Hardware used

The system consists of six sensors and devices in total, not including the server and computers in the control room and MEL:

- Arduino Uno microcontroller
- Shock/impact sensor
- ESP8266 Wi-Fi module
- Proximity sensor
- CCTV camera
- Wireless routers

IoT framework/module

The IoT framework consists of the Arduino, the ESP8266, the shock and proximity sensors. Together, they are responsible for detection of collisions in each vehicle; they are described in more detail below.

Figure 5: IoT framework

Arduino Uno **-** The Arduino Uno is a consumer grade microcontroller which is especially popular for IoT use cases. The reason for choosing this over other hardware like the Raspberry Pi is due to the lower power consumption and user-friendliness to setup. This device is responsible for controlling the shock and proximity sensors, and with the help of the ESP8266 module, collecting to the routers to broadcast accident information.

ADXL377 Shock Sensor **-** The shock detector is a device which indicates whether a physical shock or impact has occurred. This is used to detect an accident, in tandem with the proximity sensor. The ADXL377 is a robust shock sensor that is capable of detecting shocks of up to 200G.

ESP8266 Wi-Fi module **-** The ESP 8266 is a cost-effective Wi-Fi module which can be used by the Arduino to connect to wireless routers. This system uses this for the car-installed setup to connect to the MEl and send broadcast messages in the event of an accident.

M18DPO Proximity Sensor **-** This is used to detect objects which are in front of the vehicle. While special types of proximity sensors are used to detect a limited type of object, this particular use case requires detecting the proximity of metal (from the bumper of the vehicle in front), which is performed by the M18DPO, an inductive proximity sensor. Other types may be used for detecting e.g. pedestrians.

CCTV Camera **-** The CCTV camera is used for capturing video footage at the MEL. It continuously overwrites footage as to save space; when an incident occurs, the previous 5-10 minutes are saved to disk to prevent then being overwritten. This can then later be accessed by the control room, and forensics teams if required. For larger disk capacities, a longer time period can be used instead, as there is a tradeoff between video quality and storage capacity.

Software Used

The software used in implementing the system are listed below:

- Arduino IDE
- HTML
- PHP

The Arduino IDE is used in the system for developing the hardware in the car for the detection phase of the system. It is responsible for controlling the shock detector, the proximity sensor and for connecting to the routers in the MEL, and for sending broadcast messages to the router when an accident occurs.

HTML and PHP are used to create the response system for the control room and the central server: when the data is input as XML then it is converted automatically to JSON, and when the accident is detected, the central server alerts the MEL (identified by checking the sender identity) about the control rooms present, where the control room is alerted to check whether an accident has occurred, and the response systems are deployed.

The structure of the XML stored at the time of backend registration is shown below. It contains attributes for each room such as range (in kilometres), latitude, longitude and id; these are used so that filtering by the server is faster (there is no need to go to the lowest level to examine whether a control room is in a valid range of the MEL). Other details as the name of the control room and the maximum capacity are stored as values within the room XML.

```xml
<rooms>
 <room id="0" rangekm="20" lat="33.4329"
lng="39.4561">
   <name>Central Railway Control Room</name>
   <capacity>8</capacity>
 </room>
</rooms>
```

Figure 6: XML structure of control room data central server

For calculating the distance between the MEL and the control rooms near it, the haversine formula is used, due to the lower error as compared to simple Pythagorean distance. [6] The diagram below depicts the status of MELs (circles) and the control rooms (hexagons). Using the haversine formula, the distance between the MEL and the control room is calculated. In the event of more than one control room visible to the MEL, the closest one is chosen: red lines indicate rejected control rooms.

Aditya Subramanian *et al*, International Journal of Computer Science and Mobile Applications,
Vol.6 Issue. 4, April- 2018, pg. 10-18 ISSN: 2321-8363
Impact Factor: 5.515

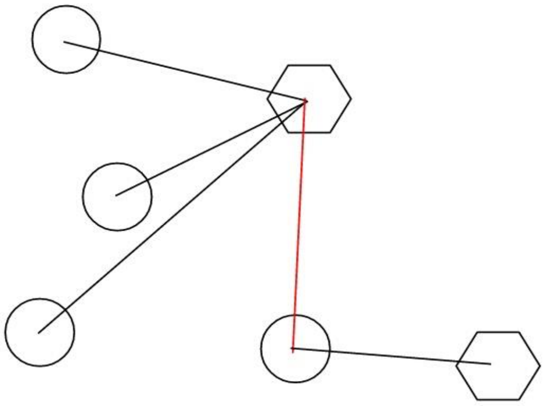

Figure 7: Distance calculation using Haversines.

6. Conclusion & Future Work

In this paper, an IoT-based framework for the detection accidents, with the aim of reducing deaths due to delayed responses, has been discussed. While the system has not been implemented yet, there are a few insights that can be drawn from the system, as week as regions of improvement. Due to the distributed nature of the detection system, it is expected that the system will hold up reasonably even under periods of stress; however, the system can be improved further: currently, the single point of failure is the central server; since all MELs in the city require access to the central server, the entire system can fail in the event that the central server becomes unavailable. Another concern is that of cost: placing routers along the road stretch will incur costs, depending the length of road and the power/range of the router.

Some of these concerns can be addressed to an extent. The central server can either be moved to a high-availability cluster hosted by a cloud service provider (which, in turn, will incur extra, possibly exorbitant, costs) or the move to a cloud pub/sub notification system can be used instead, where the control rooms receive accident data without the need of a server (which may be more cost-effective in the long run, but limits scope for future enhancements). As for the cost, buying routers in bulk can reduce the overall cost; also, the MELs can initially be introduced to accident prone areas instead of the entire city. Alternatively, the cost can be offloaded to the users, wherein cellular IoT is used to contact the central server, at the cost of omission of CCTV footage, or a hybrid solution wherein the MELs are used within areas where they are installed and using mobile communications in remote areas. If machine learning is used by the MEL's computer systems, CCTV footage can be pre-emptively sent to the control room (via the central server) if it detects an accident. [3]

The shock sensor, which is an off-the-shelf product, may not be robust enough to survive an accident. This entails the use of the proximity sensor to serve as a backup (in the event of annihilation of the shock sensor); however, the same sensor which is used for triggering the airbags can be used to trigger the system as well. This is a much better option that allows for doing away with the proximity sensor entirely; however, not all cars - most notably, older cars not up to standard and "beater cars" (cheap, poorly maintained cars intended for use as a temporary vehicle for those who cannot afford better cars) - have airbags, and installing airbags is a costly undertaking which may not come to fruition. As such, the system of choice varies on a case by case basis (e.g. luxury cars and higher-end cars can make use of the airbag-based method.) The system can be foregone altogether if the CCTV cameras are used to detect accidents using machine learning, with high probability [10].

An interesting point to note is that, as far as vehicle-to-vehicle accidents are concerned, 100% coverage of the system (user registration) is not required, as it can be simplified to the "herd immunity" problem - cars that have the system installed on their vehicle which get involved in an accident with those that do not, will still result in the control room being notified of an accident occurring, at least in the event of frontal collisions.

References

[1] SaveLIFE foundation, Road Crash Factsheet. http://savelifefoundation.org/wpcontent/uploads/2017/09/Road-Crash-Factsheet_SLF_2016.pdf

[2] E. Nasr, E. Kfoury and D. Khoury, 2016, "An IoT approach to vehicle accident detection, reporting, and navigation," *2016 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, pp. 231-236.
doi: 10.1109/IMCET.2016.7777457

[3] K. Sandeep, P. Ravikumar and S. Ranjith, 2017, "Novel Drunken Driving Detection and Prevention Models Using Internet of Things," *2017 International Conference on Recent Trends in Electrical, Electronics and Computing Technologies (ICRTEECT)*, pp. 145-149.
doi: 10.1109/ICRTEECT.2017.38

[4] Archana D., Boomija G., Manisha J. and Kalaiselvi V. K. G., 2017, "Mission on! Innovations in bike systems to provide a safe ride based on IOT," *2017 2nd International Conference on Computing and Communications Technologies (ICCCT)*, pp. 314-317.
doi: 10.1109/ICCCT2.2017.7972296

[5] Chanjin Kang and Seo Weon Heo, 2017, "Intelligent safety information gathering system using a smart blackbox," 2017 IEEE *International Conference on Consumer Electronics (ICCE)*, pp. 229-230.
doi: 10.1109/ICCE.2017.7889294

[6] A. Celesti, A. Galletta, L. Carnevale, M. Fazio, A. Ĺay-Ekuakille and M. Villari, 2017, "An IoT Cloud System for Traffic Monitoring and Vehicular Accidents Prevention Based on Mobile Sensor Data Processing," in *IEEE Sensors Journal*, vol. PP, no. 99, pp. 1-1.
doi: 10.1109/JSEN.2017.2777786

[7] Z. Chaczko, F. Jiang and B. Ahmed, 2017, "Road Vehicle Alert System Using IOT," *2017 25th International Conference on Systems Engineering (ICSEng)*, pp. 488-493.
doi: 10.1109/ICSEng.2017.72.

[8] M. A. A. Mamun, J. A. Puspo and A. K. Das, 2017, "An intelligent smartphone based approach using IoT for ensuring safe driving," *2017 International Conference on Electrical Engineering and Computer Science (ICECOS)*, pp. 217-223.
doi: 10.1109/ICECOS.2017.8167137

[9] R. Patel, V. K. Dabhi and H. B. Prajapati, 2017, "A survey on IoT based road traffic surveillance and accident detection system (A smart way to handle traffic and concerned problems)," *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, pp. 1-7.
doi: 10.1109/IPACT.2017.8245066.

[10] P. A. Nandurge and N. V. Dharwadkar, 2017, "Analyzing road accident data using machine learning paradigms," *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, pp. 604-610.
doi: 10.1109/I-SMAC.2017.8058251

[11] M. Afrin, M. R. Mahmud and M. A. Razzaque, 2015, "Real time detection of speed breakers and warning system for on-road drivers," *2015 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, pp. 495-498.
doi: 10.1109/WIECON-ECE.2015.7443976.