ISSN: 2321-2152

IJMECE

International Journal of modern electronics and communication engineering

E-Mail
editor.ijmece@gmail.com
editor@ijmece.com

www.ijmece.com

ISSN 2321-2152
www.ijmece.com
Vol 11, Issue 1, 2023

ACCIDENT DETECTION AND NOTIFICATION SYSTEM LEVERAGING AWS

Mrs.Sujatha Godavarthi[1]., K.Uma Maheshwari[2]
1 Assistant Professor, Department of CSE, Malla Reddy College of Engineering for Women., Maisammaguda., Medchal., TS, India

2, B.Tech CSE (19RG1A05k9),

Malla Reddy College of Engineering for Women., Maisammaguda., Medchal., TS, India

Abstract

With the constant advancements in the automotive industry and the increase in traffic volume, chances of fatality in road accidents have increased. This poses a dire need to develop a good and reliable solution to reduce the fatalities. One way to achieve this goal is to build a system that detects road accidents and notifies medical services as soon as possible. The aim of this research paper is to provide a robust accident detection and notification system where the data collected from the accident site is communicated to an emergency medical service provider as well as the victim's emergency contacts such as family members or close friends. The system described in this paper successfully takes values from accelerometer, detects a crash and sends a notification, a text message and an email using AWS to the concerned parties with the latency of approximately 20 seconds and this was achieved by using an internet connection with a speed of 3.76 MBPS. Since AWS is reliable, scalable and fast, the cloud infrastructure provides a better platform for the system.

I INTRODUCTION

According to WHO's Global Status report on road safety - 2018, every year, 1.35 million people lose their life in road accidents. Road traffic injuries cause considerable economic losses for the families of those involved. As per a report published by the Ministry of Road Transport and Highways (Government of India), the total number of accident - related deaths reported in the year 2019 was 1,51,113. It can be seen from the data that road accidents still remain a primary cause of death, disabilities, and hospitalization in India. India is ranked first in the number of accident-related deaths out of 199 countries and around 11% of accident-related deaths in the world happen in India.

The pie-chart shown in Fig. 1 provides details of persons killed in accidents, classified by the type of impacting vehicles. Buses, auto rikshaws, trucks and lorries etc. can be categorized as four-wheelers with a little tweak in the acceleration threshold of the algorithm. With a robust accident detection and notification system (ADNS), the number of fatalities and serious injuries can be decreased substantially. By calling and notifying as quickly as possible, such a system will be useful to the emergency centres to take action on their Medical Services. And hence, the chances of death in accidents due to

delay in medical treatment can be reduced as much as possible and the casualty rate in road accidents can be significantly decreased increases to a significantly higher point. One such system that is proposed in this work uses cloud infrastructure (AWS) and IoT technology. This system is useful in many secluded areas like mountain range, villages, interior roadways; even in night time where there's almost no-one around. In such cases, if an accident occurs due to any reason, the system could be proven life-saving. In order to reduce the fatality rate, there is a strong need for a system which is robust, generic and can be installed in any vehicle.

II LITERATURE SURVEY

"Car accident detection and notification system using Smartphone"

This report shows the outcome by applying large scale data mining techniques on the Finnish roads. From the research study it is very difficult task to perform because the collected data have uncertainty, incomplete and error values. So the data exploration is a challenging task. The data used in the process have been collected from Finnish road administration data sets. The data used in the process have been collected from Finnish road administration data sets. The main target of our project is to look into practicability of Robust clustering, to find the associations and repeated item sets and applying apprehend methods for the analysis of road accidents. While the results display the selected mining techniques and methods were capable to the understandable patterns. To calculate the accident frequency count as a parameter /c-means algorithm is used to cluster the locations. To characterize the surface conditions association rule mining is used. data mining skills disclosed different environmental reasons associated with road accidents. Intersection on highways has been identified as a dangerous for fatal accidents.

iBump: Smartphone application to detect car accidents

Traffic accidents are a fact of life. While accidents are sometimes unavoidable, studies show that the long response time required for emergency responders to arrive is a primary reason behind increased fatalities in serious accidents. One way to reduce this response time is to reduce the amount of time it takes to report an accident. Smartphones are ubiquitous and with network connectivity are perfect devices to immediately inform relevant authorities about the occurrence of an accident. This paper presents the development of a system that uses smartphones to automatically detect and report car accidents in a timely manner. Data is continuously collected from the smartphone's accelerometer and analyzed using Dynamic Time Warping (DTW) to determine the severity of the accident, reduce false positives and to notify first responders of the accident location and

owner's medical information. In addition, accidents can be viewed on the Smartphone over the Internet offering instant and reliable access to the information concerning the accident. By implementing this application and adding a notification system, the response time required to notify emergency responders of traffic accidents can reduce the response time and perhaps help in reducing fatalities.

"Mobile application for automatic accident detection and multimodal alert."

This paper presents HDy Copilot, an Android application for accident detection integrated with multimodal alert dissemination, both via eCall and IEEE 802.11p. The proposed accident detection algorithm receives inputs from the vehicle, via ODB-II, and from the Smartphone sensors, namely the accelerometer, the magnetometer and the gyroscope. The Android Smartphone is also used as human machine interface, so that the driver can configure the application, receive road hazard warnings issued by other vehicles in the vicinity and cancel countdown procedures upon false accident detection. A prototype implementation was validated via laboratory tests.

III EXISTING SYSTEM

- There are several other research papers published on this topic. The algorithm mentioned in the current paper was tested on data from NHSTA, USA's crash- data for accuracy of accident severity, which is mentioned in the later part. The work published in utilized Android smartphone sensors for car crash detection.

- Ahirrao et al suggested a separate collision detection circuit installed in the car and a separate notification system using Android App. The circuitry utilizes LDR and photodiode as the sensors for collision detection.

- The work presented in and explains an accident detection and notification system using an accelerometer and Raspberry Pi.

- Various approaches can be seen in research papers published on this topic recently. One of the solutions is e-Notify which aids in detecting and notifying traffic accidents. An onboard unit (OBU) is required for this system. This can be costly and it's difficult to equip every vehicle with an OBU.

- There is also an eCall system developed by The European Commission and its deployment is, by law, necessary in all vehicles that are manufactured after the year 2015. This system detects an accident and then the emergency services like 112 (or 999 in UK) are informed.

Limitations Of Existing System

- costly and it's difficult to equip every vehicle
- No Information available of riders or drivers
- Not able to notify beloved ones
- Requires Police action for Information which will be time consuming

ISSN 2321-2152
www.ijmece.com
Vol 11, Issue 1, 2023

IV PROPOSED SYSTEM

- The system described in this paper successfully sends notifications, a text message and an email using AWS to the concerned parties with the latency of approximately 20 seconds and this was achieved by using an internet connection with a speed of 3.76 MBPS. Since AWS is reliable, scalable and fast, the cloud infrastructure provides a better platform for the system .

- Our system uses cloud infrastructure (AWS). This system is useful in many secluded areas like mountain range, villages, interior roadways; even in night time where there's almost no-one around. In such cases, if an accident occurs due to any reason, the system could be proven life-saving.

- In order to reduce the fatality rate, there is a strong need for a system which is robust, generic and can be installed in any vehicle.

- A simple web application is developed. This application is useful for receiving push notifications i.e., "An accident is detected!!!" whenever the algorithm triggers AWS SNS. The app receives notifications using AWS SNS. More features like displaying details of accident and owner can be added.

- A website is developed using HTML, CSS, JavaScript. The purpose of this website is preliminary to be the software at the hospitals and the medical centers. The website shows the last accident's details as well as all the records of the accidents that have happened. Entire system is implemented using python with Django framework and MySQL as database.

V IMPLEMENTATION

Calibrating the Accelerometer

(MPU-6050) MPU-6050 needs to be calibrated first since every module will have different off-set errors and they need to be checked and edited in the source code accordingly for every system that is developed. These offset-errors are present in the accelerometers due to misalignment caused during manufacturing process. For that, the accelerometer needs to be placed on a flat surface and measure the acceleration values in each direction i.e., x-axis, y-axis and z-axis. The acceleration values for x-axis, y-axis and z-axis ideally should be 0g, 0g and 1g respectively. But practically, there would be offset errors and they need to be added or subtracted accordingly to get the desired output with minimized error.

Accident Detection and Severity Algorithm

The flowchart of the basic working of the algorithm. If the acceleration/deceleration value exceeds 5g (where g=9.8 m/s2) then it needs to be considered as an accident [22]. To take an estimation of the severity, the acceleration values can be broken down as follows: if the acceleration is between 5g and 13g, it's considered High and for 13g and above, it's considered extreme. The acceleration values can directly be obtained by the accelerometer i.e., MPU6050. However, to estimate the severity,

the change in velocity is also required. The duration for which deceleration exists determines the change in velocity. The longer the duration, the larger the change in velocity i.e., □v or DeltaV. Just like acceleration, there are two ranges of DeltaV to estimate the severity. If the DeltaV ranges from 12.8 kmph to 22.5 kmph, it's considered High and above 22.5 kmph, extreme. These velocities can be obtained by performing integration on the acceleration values. One such approach is to pass the acceleration values through a Butterworth Low-Pass Filter of order '2'. MPU-6050 has such a filter inbuilt. So, the values obtained thus are the filtered values and they can be directly integrated to get values of DeltaV (□v) and now steps can be taken to estimate severity. The pseudo code for detection and severity ***estimation is mentioned below***:

1. Get acceleration/deceleration values.
2. Integrate values and calculate velocity.
3. Get Pi = magnitude of maximum deceleration in i th direction. (i = x,y)

ADS Hardware

The most important part of the system is detection and computation of severity. AWS Command Line Interface (AWS CLI) is installed on the Raspberry Pi's Raspbian operating system so that AWS can be accessed and the credentials can be verified. The algorithm is implemented in Python3 and will run locally on Raspberry Pi. The algorithm will constantly run on the system and the accelerometer (MPU-6050) will measure acceleration values and the algorithm along with pressure sensor will detect if there was an accident and then start buzzer and send notifications accordingly. Since the computation for severity and detection runs locally on Raspberry Pi, it takes about 6-8 seconds to do it (considering the 3b model). When the algorithm detects an accident, AWS services i.e., SNS, DynamoDB and PinPoint are triggered. All of which use boto3 client which is a library to help us integrate AWS functionalities locally in the main program. Boto3 clients fetch AWS credentials from the installed AWS CLI or they can be passed as parameters manually. In addition to this, additional parameters like weight of the vehicle could be added. Also, the system can be customised according to the different types of terrains like slippery, rocky etc. However, that would require testing in a standard facility with different real vehicles

AWS Credentials, DynamoDB, SNS and PinPoint

AWS Command Line Interface is installed on the Raspberry Pi so that basic AWS commands can be run locally as well since important and essential credentials can be set in a separate hidden file which AWS client would read from. These credentials contain three details: 'AccessKeyID', 'SecretAccessKey' and 'Region'. Each of these is different for different users and need to be configured while developing the system to maintain reliability and security. Setting up AWS cloud credentials

needs to be done for every user. AWS DynamoDB provides a scalable and easy to use NoSQL database. This database has two tables named Accidents and Vehicle. The Accidents table contains 'AccTime', 'Impact Area', 'Impact Type', 'Overall Severity' and 'OwnerID'. The Vehicle table contains 'OwnerID', 'Name', 'Vehicle', 'Plate No.', 'Insurance ID' and 'License No.'. The Accidents table is triggered when the algorithm detects the accident and the data computed by the algorithm is sent to the appropriate field in the table accordingly. The Vehicle table is also triggered at the same time and is used to show the owner's details in the notification in the Android app, email and website. AWS SNS (Simple Notification Service) comes handy when you want to send a notification to the user on their phone. When the accident is detected, the boto3 client of SNS is invoked and two types of notifications are sent. One using the Firebase Cloud Messaging and the android app as push notification which basically shows that an accident is detected with severity. The second notification goes as an email containing the accident details and the owner's details i.e., all the fields from both the DynamoDB tables - Accidents and Vehicle. AWS PinPoint is a service to send voice messages. Along with the push notification, a voice-note saying "An accident has occurred" is sent to the contacts.

Android Application and Website

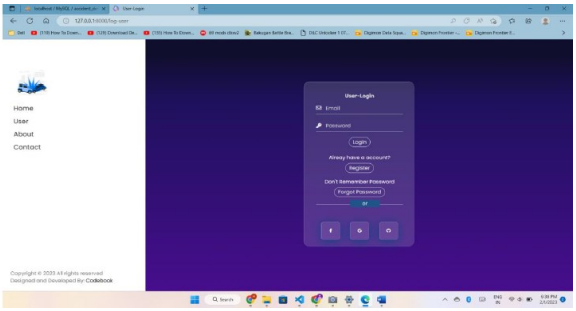
A simple android application is developed. This application is useful for receiving push notifications i.e., "An accident is detected!!!" whenever the algorithm triggers AWS SNS. The app receives notifications using AWS SNS and FCM. More features like displaying details of accident and owner can be added. A website is developed using HTML, CSS, JavaScript. The purpose of this website is preliminary to be the software at the hospitals and the medical centers. The website shows the last accident's details as well as all the records of the accidents that have happened. Entire system is implemented using various technologies; references of which you can find from . The GitHub Repository for the code is mentioned at [40]. To test the algorithm, experimentations were performed on these two systems of trolley-carts. A series of experiments were performed to make accident-detection possible. The accelerometer and pressure sensor are interfaced to the Raspberry Pi and the system is rigidly tied to the trolley. The algorithm in Raspberry Pi takes real time data from the accelerometer and sends the output to the assigned output device. Different accident scenarios were replicated on these prototype models. After 7-8 seconds of one collision, the accident is detected and a notification, a text message, an email and a voice note are received through AWS showing the impact area and severity. The results are shown in the following section. However, the proposed work is to be extended by including camera, microphone, GSM module, some more AWS services and

ISSN 2321-2152
www.ijmece.com
Vol 11, Issue 1, 2023

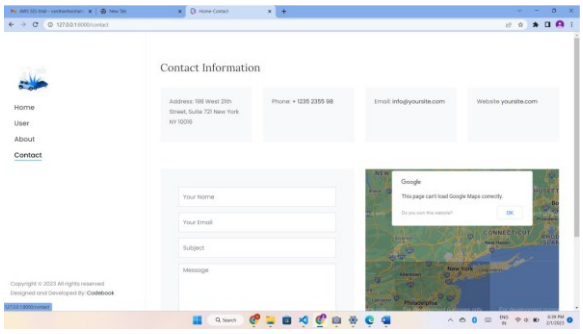
enhanced web application and android application.

VI RESULTS

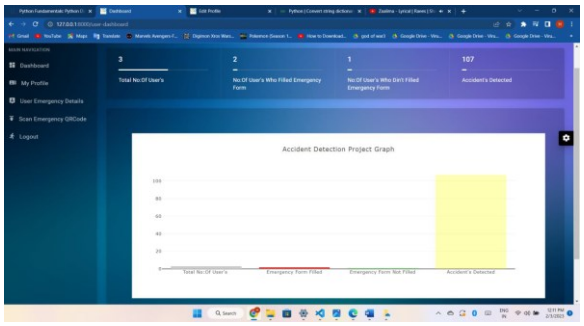
Home Page



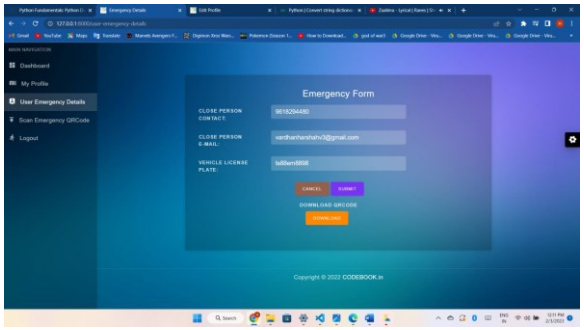
User Login



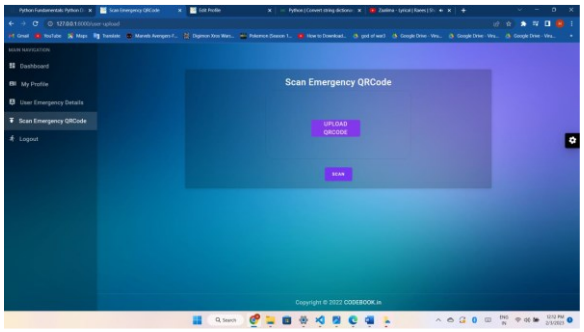
Contact Information



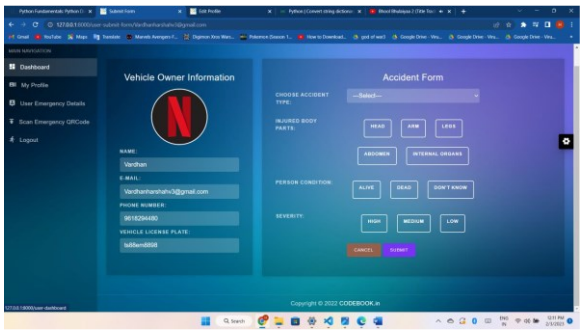
Analysis



Emergency form



Scan Emergency QRCode



Accidental Form

VII CONCLUSION

The increasing transportations are causing more and more road accidents resulting into people's deaths. It is crucial to get them immediate medical attention and the system described in this work provides just that. AWS is a reliable, scalable and easily maintainable cloud service. Also, since the sensors are rigidly mounted to the vehicle's body, the reliability isn't compromised, compared to the approaches involving smartphone sensors. Apart from that, unlike other approaches, the proposed model successfully estimates severity as well. As mentioned in the Results and Analysis section, the system successfully detects an accident and notifies the concerned within approximately 20 seconds and medical treatment can be started as soon as possible and hence, it increases chances of survival. The system was tested with internet speed of 3.76 MBPS; but this latency depends on factors like network connectivity, network-traffic etc. and may take longer to notify.

REFERENCES

[1] Road Traffic Injuries, Factsheets, WHO. [Online] Available: https://www.who.int/news-room/fact-sheets/detail/roadtraffic-injuries

[2] Road Accidents in India 2019, Ministry of Road Transport and Highways. (pp. 11-12,27-28)[Online]
Available:
https://morth.nic.in/sites/default/files/RA_Uploading.pdf

[3] NHSTA Crash Data. [Online] Available: https://www-nrd.nhtsa.dot.gov/database/veh/

[4] H. M. Ali, S. Alwan. "Car accident detection and notification system using smartphone". Saarbrucken: LAP LAMBERT Academic Publishing, 2017.

[5] F. Aloul, I. Zualkernan, R. Salma, H. Ali, M. Merri. "iBump: Smartphone application to detect car accidents." Computers & Electrical Engineering 43, 2015 (pp: 66-75).

[6] B. Fernandes, V. Gomes, J. Ferreira, A. Oliveira. "Mobile application for automatic accident detection and multimodal alert." In 2015 IEEE 81st Vehicular Technology Conference (VTC Spring), (pp. 1–5), May 2015.

[7] B.K. Dar, M.A. Shah, S.U. Islam, C. Maple, S. Mussadiq, S. Khan "Delay-Aware Accident Detection and Response System Using Fog Computing." IEEE Access. 2019 May 1.

[8] S. M. Ahirrao , P. Dhanrale, L. Mahant, H. Kotwal. "Accident Detection and Notification System Using Android." International Journal on Recent and Innovation Trends in Computing and Communication 3, no. 3: (pp:1084-1086))

[9] V. Reddy, L. Sree, V. Kumar. "Design and development of accelerometer-based system for driver safety." International Journal of Science, Engineering and Technology Research (IJSETR), (pp:3463– 3468), 2014.

[10] A. Shaik et al., "Smart Car: An IoT Based Accident Detection System," 2018 IEEE Global Conference on Internet of Things (GCIoT), 2018, pp. 1-5, doi: 10.1109/GCIoT.2018.8620131.