Development Of A Mobile-Based Accident Detection And Notification System With Multi-Modal Data

Lead Author

Micheal Damilola Alu

Affiliation:

Department of computer science Redeemer's University Ede, Osun State

Road accidents is one of the factors responsible for fatalities and lifelong disabilities globally. The predominant increase in mortality rate on the highway has been attributed to the late arrival of emergency authorities, which could be because of delays in reporting time. The literature has suggested a number of approaches to deal with this significant issue. Some of these approaches include the use of machine learning algorithms. However, many of these works have been limited to single-mode (video, audio, or image) accident detection methods. In addition, some of the existing study explore the use of intelligence transportation system, which could be considered expensive to use. This research has developed a mobile-based accident detection and notification system with multi-modal data. The detection system has incorporated trained simulated data that is validated by unseen data online. The simulated data was characterized by some selected modal data (accelerometer, sound, and gravitational force magnitude) equivalent to the online features. The simulated data was trained using the multilayered perceptron artificial neural network (ANN) model. The trained model was tested using online data from the data world platform. The observation with the simulated data showed that the model achieved an accuracy of 99.5%. The result of the experiment on the online data showed 99.8% accuracy. At the end of the modelling, the ANN model was integrated into an android mobile-based accident detection and notification system. Furthermore, the system was tested with several case study and the result showed that the system performed as expected.

Keywords: accelerometer, accident, artificial neural network, audio, G-force, android mobile application.

Co-Authors: Adewale Opeoluwa Ogunde; Mba Obasi Odim; Toluwase Ayobami Olowookere Department of Computer Science, Redeemer's University, Osun State, Nigeria

Introduction

One of the factors responsible for fatalities and lifelong disabilities globally is road accidents. According to the 2015 Global Status Report on Road Safety from the World Health Organization, 26.7 per 100,000 people died in traffic accidents in Africa in 2013 [1]. Over 1,300,000 people are killed annually in road accidents [2]. Some of the victims died immediately after the collision, while others died later. This could be due to late arrival of emergency service' agencies or delay in reporting the incident. These problems may be resulting from a lack of an appropriate mechanism to detect and communicate the occurrence. Automated accident detection and notification systems are valuable tools to help in the early detection and reporting of accidents. Numerous techniques have been used in this regard, including deep learning, and some machine learning algorithms that could use single or multimodal data for detection.

Machine learning techniques are mathematical model mapping ways for learning or uncovering unseen patterns in data [3]. Deep learning (DL) is a computer-based modeling approach consisting of several processing layers used to analyze data representation at different levels of abstraction [4]. It is one of the machine learning approaches centered on extracting features from raw data by utilizing numerous layers to recognize various input data elements. It aids in modeling data for prediction, classification, and other functions [5]. Deep learning algorithms come in a variety of forms, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory Networks (LSTM), Artificial Neural Network, etc. In this research, Artificial Neural Network (ANN) will be used to model some mobile device built-in sensors data, referred to as modal data.

Multi-modal data refers to the collection of mobile devices' built-in sensor data; the generation of these data results from changes in the environment. Over the last decade, general user activities such as sitting, walking, jogging, running, or walking up and downstairs have been classified using smartphone sensors [6]. Some of the modal data generated by a mobile device include accelerometer, gravitational force, pressure, sound, temperature, etc. Modal data has contributed immensely to mobile application-based detection research. However, with the deep learning approach, sensors' data may be processed using different techniques to infer hidden information. The primary goal in this research was to model and develop a mobile-based application to detect the occurrence of an accident, and to send a notification to the appropriate authority or contact person. The widespread mobile device has allowed Short Message Service (SMS) to provide some innovative ideas for accident notification, including the ability to send a message even when there is no available data subscription on a mobile device. In this research, SMS was considered for offline notification to the contact persons. Another form of a report adopted in this research was the Firebase Cloud Messaging (FCM) Push Notification system. FCM is a popular client/server communication option for mobile devices. FCM would help notify the emergency authority in real-time, reducing the delay in the report. Both methods of notification disclosed the necessary information regarding the occurrence of the accident.

The research has suggested a number of strategies to deal with this important issue. However, some of these methods were limited to single-mode input detection methods, i.e., using only video, image, or audio to detect an accident [7] [8] and few users (e.g., vehicle drivers) [9]. Some researchers suggested using deep learning to reduce false-positive occurrences [7] [10]. Others have tried to introduce machine learning methods like Support Vector Machine (SVM) and Random Forest (R.F.) [5] [11] [12] the research attempts were limited to road traffic crash images and videos classification. This study seeks to develop an accident detection and notification system that inputs mobile device sensor data (audio data, accelerometer data, and gravitational force) into artificial neural networks to classify an accident occurrence. In addition, the goal is to reduce false-positive, increase accuracy, and allow a large audience to use the system with their android mobile phone.

Related Works

In section, we have reviewed several studies relating to the development of an accident detection system.

[13]addressed an accident detection system that occurs due to the carelessness of the person driving the vehicle. An accident warning system that warns the driver of the vehicle was developed as a result of the research. Accidents happen when a driver is not in a position to control the car. The research adopted the use of AT Mega 328P microcontroller Arduino and GSM Module for accident alert, and the work also introduced a notification delay which reduces the false

positive compared to previous work. The researcher recommended introducing an effective machine learning technique for detecting accident events.

According to [11], vehicles and their conditions were examined to come to a conclusion about a possible accident. The dataset consists of 800 photos, of which 200 were used for training, and 200 were used for testing and verifying. There were 400 images for a light collision and 400 for a severe impact. The program performed with a 93 percent accuracy rate. The result showed that clustering algorithms could successfully detect accidents, But the research entirely depends on the network for notification, and the classification is limited to videos and images.

In the work of [8], the Gaussian mixture model (GMM) and CNN was used to develop an optimized frame detection in vehicle accidents. The result of the research was broken down into two phases; 1) The number of frames in the dataset was reduced by about 51% with the use of GMM, while for phase, 2) The prediction accuracy with CNN was found to be 85 percent on the local dataset, 85 percent on the SDD Dataset, and 86 percent on the VMMRdb Dataset. The major limitation here is that the research only relies on video data for training the model.

[14] developed an ensemble deep learning, and multi-modal data from dashboard cameras for detecting car crashes. The research builds a car crash detection system using deep learning methods, gated recurrent units (GRU), and a convolutional neural network (CNN). The ensemble method employed was a weighted average ensemble. The suggested car crash detection system was validated by comparing it to single classifiers that only use audio or video data. The proposed system was based on multiple classifiers that used video and audio data from dashboard cameras. The research suggested using 3-dimension for better performance.

[9] worked on efficient accident detection and notification system. The work aimed to reduce the number of unintended deaths by developing a low-cost accident detection system. It was proposed that the research distinguishes the accident and sends a message of the accident area to worried relatives via Short Message Services (SMS) with accurate GPS location. The researchers made use of an Arduino board with a push-button installed in a vehicle and with a driver's mobile device for location information. The results of the experiments disclosed that the proposed system performed as

intended, although only limited users can use the system (limited to only vehicle drivers).

[15] also worked on the survey of accident detection systems. The research recommended using the R-CNN base model, as it provides better performance than other algorithms like Mask R-CNN, Sobel Edge, HDNN, and AdaBoost. The study only detects accident events, and it doesn't cover notification of accident events.

Based on the research carried out by [16], a feature fusion-based deep learning framework, neural network (ResNet), Spatio-temporal feature fusion model, and Conv-LSTM (Convolutional Long Short-Term Memory) was used for crash detection in video data. On the test dataset, the study's accuracy was 87.78 percent, and its detection speed (FPS > 30 with GTX 1060) was adequate. The authors clearly stated that the research was carried out in an urban environment, using a newly supplied traffic video.

In the research by [17], the detection method adopted used a GSM modem to detect accidents using an accelerometer, pressure, and noise sensor. The multiple sensors helped the system increase accuracy compared to previous work with a single sensor.

Methodology

This section provides a systematic report of the accident detection and notification system development lifecycle. The waterfall software development lifecycle (SDLC) model was adopted for the development processes. The waterfall model is quick and straightforward to accomplish and with a functional project structure. Also, it is appropriate for small-to-midsize projects and easy to evaluate and analyze the feature realization. The subsequent sub-sections describe how the adopted software development lifecycle model was used as a methodology to accomplish the aim and the objectives of this study. Section 3.1 describes the requirement gathering and analysis, section 3.2 describes the system design, section 3.3 describes the ANN model, and section 3.4 illustrates the model and system evaluation approach.

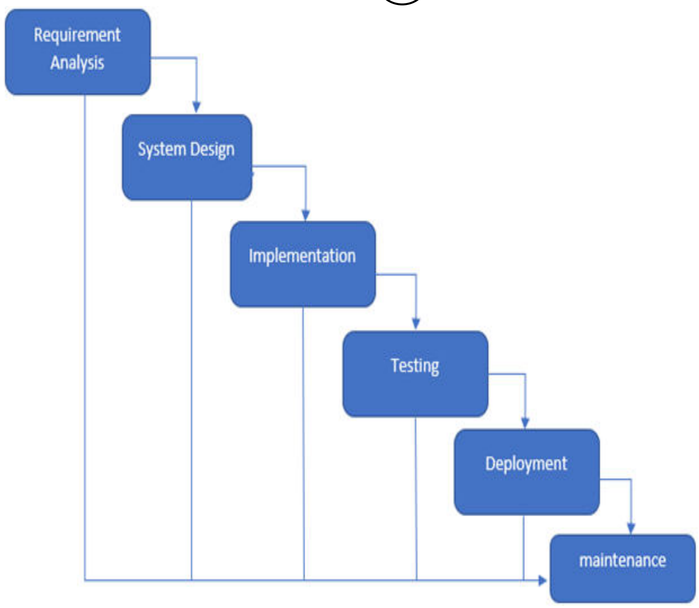


Figure 1: The waterfall software development life cycle model

3.1 Requirement Gathering and Analysis

The requirement can be broadly categorized into functional and non-functional requirements. The functional requirements for a system portray what the system is expected to do. These requirements rely upon the nature of the software system being realized, the product's presumed client, and the organization's overall processes when gathering or composing requirements. The functional requirement of the system under study is to detect accident occurrences and notify the concerned entity (emergency authority or contact person) using a mobile device with some modal data. 3.1.1 Data Collection and Description

There are two sources from which the data used in this study was gathered. They include an online repository and mobile device simulation.

The online data was downloaded from the data world research platform (https://data.world/smartcolumbusos/24b0ae75-7235-4ee6-b034-5b6ca1986733). This dataset identifies the in-vehicle monitoring data from Honda pilot vehicles captured on May 4th, 7th, and 8th of

2018. The dataset includes the 3-axes (x, y, z). The x, y, and z coordinates represent the direction and position of the vehicle at which acceleration occurred, the gravitational force magnitude (G), and the accident is labeled 0 or 1. The data collected are characterized by the following in Table 1.

Table 1: Characterized Model Input

Data	Value	Unit
1. Accelerometer	x, y, z	m/s^2
2. Gravitation Force (G-force)	G	m/s^2
3. Audio	d	decibel
4. Accident	{0,1}	-

The simulations were carried out on an Android 7.0 (API level 24) device with a CPU speed range (604MHz to 1300MHz) using a mobile application software called andro Sensor (https://play.google.com/store/search?q=androSensor&c=apps). The simulation exercise was done with an average of 300 iterations with random actions or scenarios, including rotation, shakes, throw, bounce, and the sound pitch range of 20decibel-130decibel. The result of the simulation exercise was characterized by the accelerometer, gravitational force, and audio data with an accident label. The accelerometer was found to be in a good functional state when left flat on a horizontal flat surface. The x and y values fluctuated close to 0, while the z was around $9.81m/s^2$. The vibration measurements were collected every 60 seconds for a total of 360 seconds. Thus, in total, 1700 vibration records were compiled from the simulation exercise. The simulated data collected were stored in the common separated value (CSV) format.

3.1.2 Data Preprocessing

Data preprocessing was employed to clean the data and make it suitable for the machine learning models, which also enhances the model's accuracy and efficiency. The online repository dataset was processed to handle missing data. However, the result of the process shows that there were no missing data in the online dataset; therefore, Cross-validation was used on the pre-processed data.

3.2 System Design

3.2.1 System Model Architecture

The architectural design illustrated in Figure 2 describes the relationship between the component of the system to be implemented.

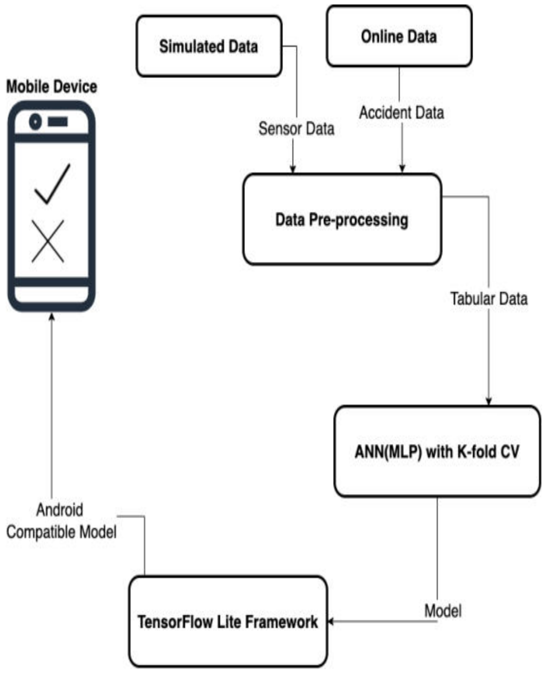


Figure 2: Accident Detection System Model Architecture

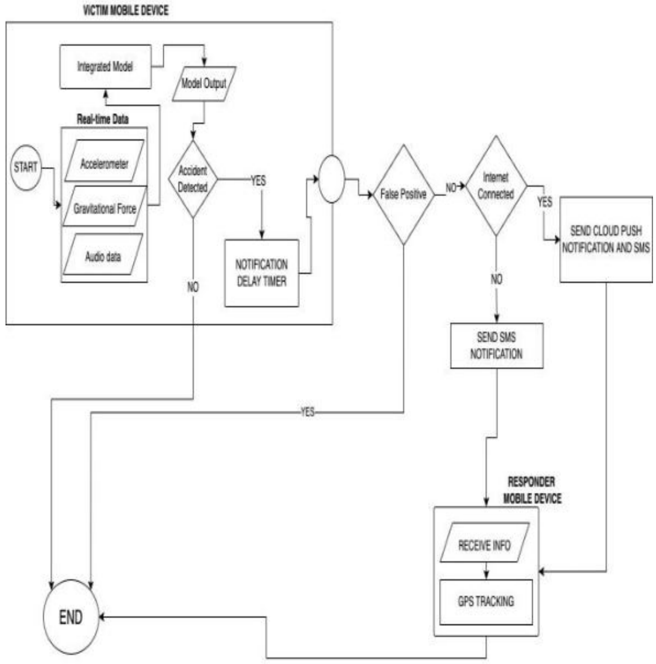
According to our architecture, the data received from both sources were characterized by three sensor data (sound, accelerometer, and gravitational force). It was observed that the online acceleration due to gravity or G-force was calculated as magnitude, i.e., in scalar value. Therefore, the gravitational force in simulated data was converted to its magnitude form using the Pythagoras equation.

$$G = \frac{\sqrt{x^2+y^2+z^2}}{9.81} \quad (1)$$

Where the *x,y, and* z represent the axes of the acceleration due to gravity at a given time and gravitational constant (9.81). The resulting dataset was partitioned using the k-fold cross-validation method for the ANN training and testing. The fundamental purpose of the model selection technique known as cross-validation is to choose hyperparameters. K-fold cross-validation was applied in this study. The number of divisions created by the cross-validation method is equal to the value entered in the numberOfFolds argument of the Cross-validation function. Six folds were used in our case since the features are (accelerometer for x axis, y axis, z axis, magnitude of gravitational force (G-force), audio value, and accident labelled). Each division produces a TrainTestData object. The output model from the training was converted to an android application compatible model, and then the compatible model from the TensorFlow lite framework was collected and implemented using an android native framework.

3.2.2 Accident Detection and Notification System Process Diagram

The system process diagram for detection and notification handling is shown in Figure 3 and explained. The mobile application device generates the selected modal data in real-time. This data was passed into the TensorFlow transformed model for detection. If an accident is detected, a notification delay timer is triggered; else, the system does not trigger a notification delay timer. The notification delay timer is used for detection control; this means, if peradventure the system detection is false positive, the user cancels the timer, else an automatic message is sent to emergency authority and contact person through SMS or real-time Push notification depending on the availability of internet connection. The Emergency authority can track the victim's location based on the location coordinate sent along with the victim's information.



3.2.3 Accident Detection and Notification System Use Case Diagram

The use-case diagram for the system is shown in Figure 4 and described in Table 2.

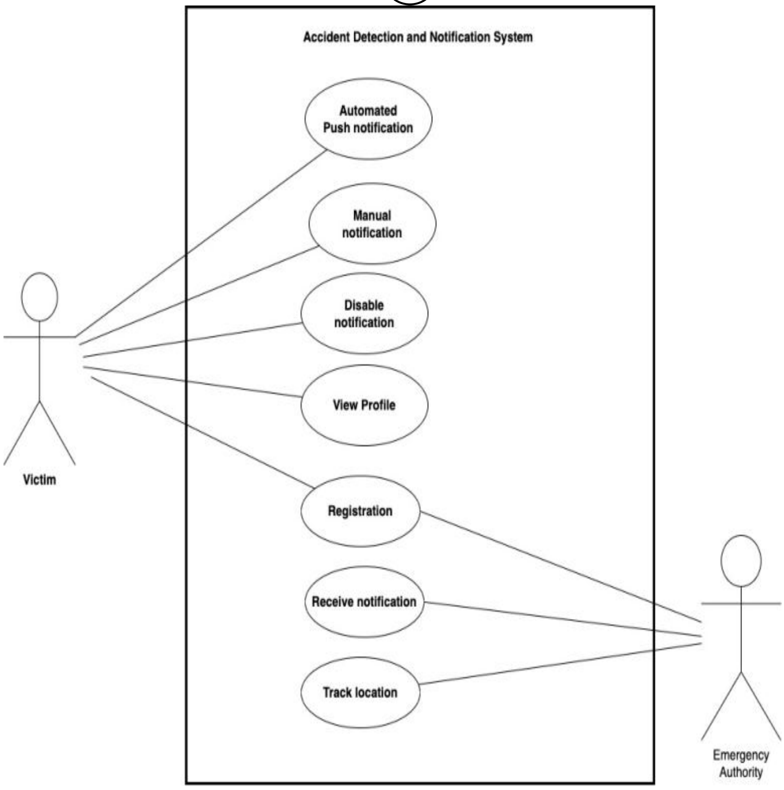


Figure 3: Use case diagram for Accident detection and notification system

Table 2: Description of the use case diagram

Actor	Goal	Input and Output
Victim	Send a manual push notification, or disable the notification.	i. Email, name, phone number, address, contact person's phone number. ii. Notification content: User information and location coordinate.
Emergency authority	Receive a notification, and track location	i. Email, organization name, phone number, office address. ii. Notification content: User information and location coordinate.

Corpus Intellectual
ISSN PRINT 2811-3187 ONLINE 2811-3209 Volume 3 NO 3 2024 Conf. Edition

3.2.4 Accident Detection and Notification System Activity Diagram

The activity diagram that emphasizes the dynamic behavior within the actors and the system is illustrated in Figure 5.

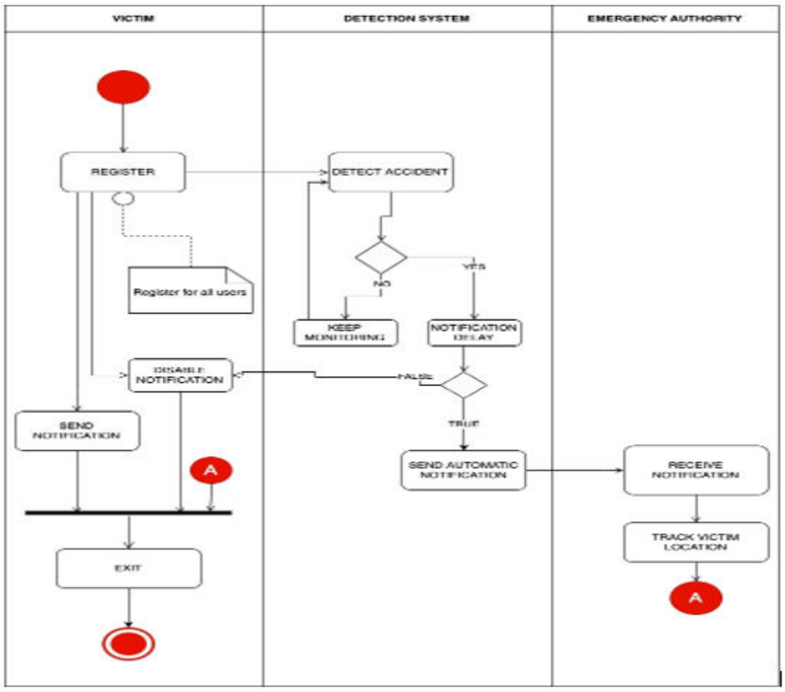


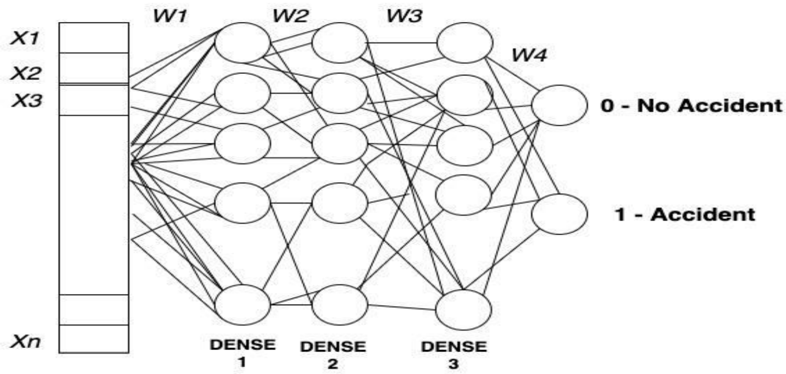
Figure 4: Accident Detection and Notification System Activity Diagram

As illustrated in the activity diagram, the system is divided into three sections (The victim, the system, and the emergency responder). Every system user registers to use the system. At this point, the basic information (email, name, organization name, phone number, address, office address) about the victim and the emergency authority are collected. After user (The victim and the emergency authority) registration, the system access was subsequently granted for accident detection monitoring on the victim's mobile device. If the detection result is "true," the notification delay timer is triggered for user control within a predefined time. Suppose the delay timer is completed without the interference of the user. In that case, the system sends an automatic push notification or SMS to any registered emergency authority(the decision to use SMS or push notification depends on the state of the network, if available, the system uses the push notification, otherwise, the SMS module is used. The location coordinate of the accident event is also sent along with the notification. The emergency authority can track the location with the provided coordinate. If the result of any detection instance is "no," the system keeps monitoring, while the victim can also initiate a direct notification without the detection system.

3.3 Artificial Neural Network (ANN) Model Description

A fully connected multilayered artificial neural network model was used to model the online and simulated dataset, the tabular data (input) was split into a training and testing set using the cross-validation method, with an input dimension of 5, and the resulting data were sequentially built with three fully connected layers. We also include the ReLU activation function, which is necessary to bring non-linearity to the model, in addition to the fully connected layer. To minimize the prediction error, the Adam optimizer was used. See Figure 6.

Figure 5: Accident detection using ANN Model



3.4 Model and System Evaluation Approach

The model was evaluated with a confusion matrix. A 2 X 2-dimensional matrix was used because the study was based on binary classification. Table 3 shows the representation of the confusion matrix. The two classes were 0 and 1, implying negative (no accident) and positive (accident) results, respectively. The diagonal values represent accurate predictions, while the non-diagonal values indicate inaccurate predictions.

Table 3: Confusion matrix

Predicted Class				
Actual Class		YES	NO	Total N P+N
	YES	TP	FN	
	NO	FP	TN	
	Total	P	N	

Terminologies of confusion matrix as follows:

True Positives [TP]: These are the classifier's positive cases adequately classified.

True Negatives [TN]: These are the classifier's negative cases adequately classified.

False Positives [FP]: These are the negative cases wrongly classified as positive.

False Negatives [FN]: These are the positive cases that were wrongly classified as unfavorable.

A confusion matrix was constructed to compute the following performance metrics:

Accuracy: The rate of accuracy was computed using the formula below:

$$Accuracy = \frac{TP+FP}{TP+TN+FP+FN} \quad (2)$$

Precision: This is referred to as positive predictive values. It was calculated by using this formula:

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

Recall: This is also referred to as sensitivity; it was calculated by using this formula:

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

All the segments developed in the implementation phase were combined into a whole framework for evaluating each unit. The overall system was tested.

Four case studies were employed in testing the system:

Case 1: The system was tested with only accelerometer data while setting the other features

(G-force and audio data) to 0 and evaluating the log result.

Case 2: The system was tested with only accelerometer data and gravitation force magnitude while setting the audio data to 0 and evaluating the log result.

Case 3: The system was tested with all features (accelerometer, G-force, and audio data) and compared the log result with other case studies.

Case 4: The system was tested with instances of simulated data and online data.

Result and Discussion

In this section, we describe the steps taken to achieve the study's results by looking into the following: A brief on the modelling procedure, the performance metrics for the accident detection model, the model integration into android device using the TensorFlow lite framework, the system evaluation, and comparison with existing studies in the problem domain. 4.1 ANN with 6-fold Cross Validation

The ANN was trained and tested on the original tabular data (simulated data) using a 6-fold cross-validation method. The same model was used on the online data. The iteration process was equivalent to the number of splits. In the case of the ANN model, the number feature size is only required. Therefore, the training and testing were in six iterations. Each iteration was required to run for 150 epochs, with 10 batch sizes each.

At the end of the experiment, the simulated model was used to predict the online data and vice versa to test the validity of the simulated data. The experimentation report was gathered as follows.
ANN Experiment with the simulated data

As shown in Figure 7, on the confusion matrix, it was observed that the false positive and true negative were 0 and 10, respectively, while the model achieves a true positive of 1059 and a true negative of 631.

Figure 6: Confusion Matrix for the ANN model on the simulated data

Figure 8 disclose the receiver operating curve (ROC) for the ANN model experiment on simulated data. According to the graph, the model mean ROC is a little away from 1.0, while each portion of training and testing disclosed an area under curve (AUC) of 1, which is much more acceptable.

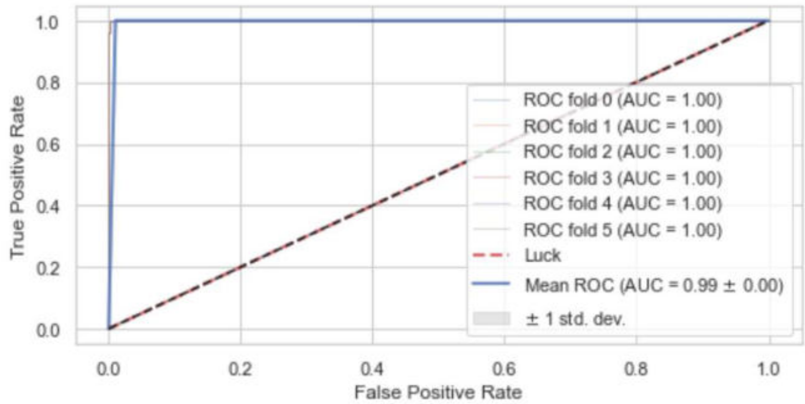


Figure 7: *The ROC curve for the ANN model experiment on the simulated data*

In Figure 9, it turns out that the AUC mean for the model is low (0.37) compared to the 0.5 average. This must have been affected by the number of false negatives (i.e., no accident event) predictions

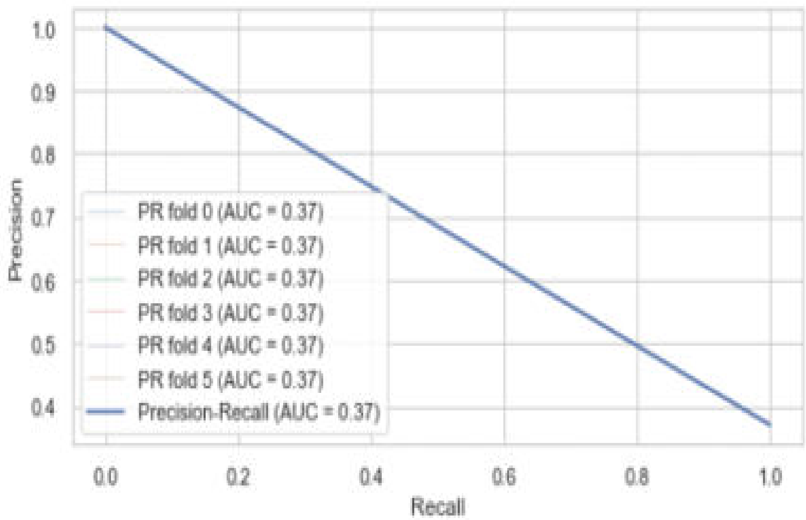


Table 4 discloses the classification metrics for the ANN model on the simulated dataset. According to the metrics, the model achieved an approximated accuracy of 99.5%, recall of 100% and the precision of 98.6% respectively.

Table 4: Classification metrics for the ANN model on the simulated data

Classification Metric	Score Value (%)
Accuracy	99.4706
Recall	100.0000
Precision	98.5938

ANN Experiment with the online data

The online data was equally trained and tested, and it is subsequently reported as follows.

Figure 10 shows the confusion matrix. According to the confusion matrix, it was observed that the false positive and true negative were 30 and 0, respectively, while the model achieved a true positive of 4661 and a true negative of 309.

Figure 8: Confusion Matrix for the ANN model on the online data

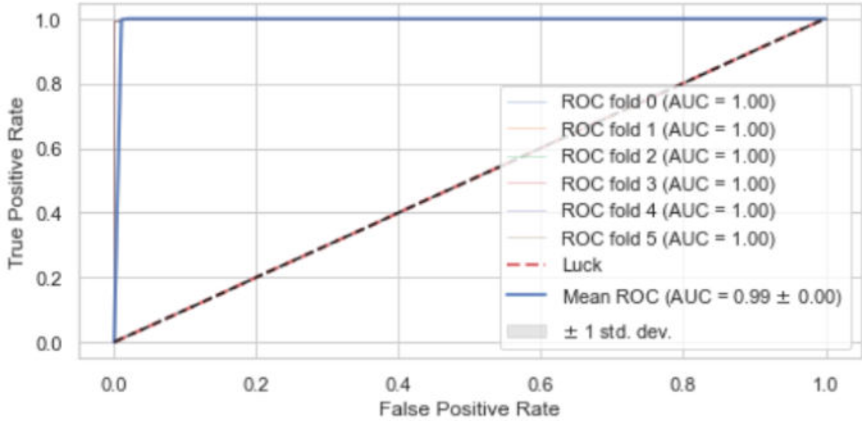


Figure 9: The ROC curve for the ANN model on the online data

In Figure 12, it turns out that the AUC mean for the ANN model is very low (0.07) compared to the 0.5 average, this must have been affected by the number of false positive predictions.

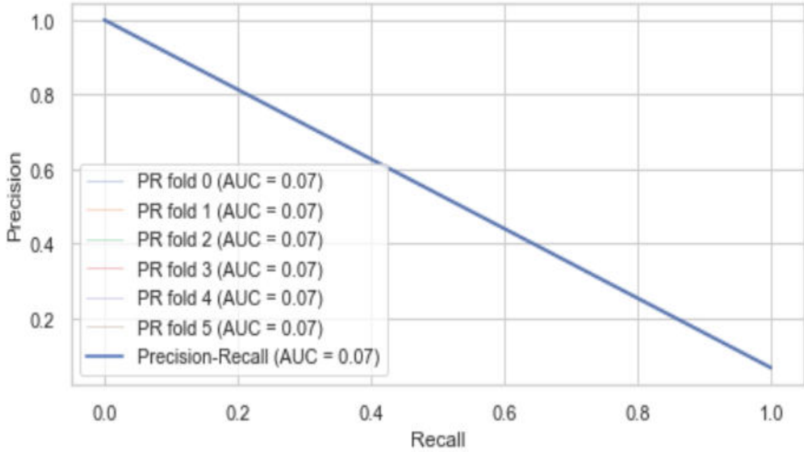


Figure 10: The PR curve for the ANN model on the online data

Table 5 discloses the classification metrics for the ANN model on the online dataset. According to the metrics, the model achieved an approximated accuracy of 99.8%, recall of 96.8% and the precision of 100% respectively.

Table 5: Classification metrics for the ANN model on the online data

Classification Metric	Score Value (%)
Precision	100.0000
Recall	96.7552
Accuracy	99.7800

To validate the performance of the simulated data, the model was saved and reloaded to predict the online dataset. It was seen that the trained model achieved an accuracy of 99.68% (See Table 6)

Table 6: Online data prediction with the ANN model for simulated data

Classification Metric	Score Value (%)
Precision	100.0000
Recall	95.2802
Accuracy	99.6800

To check the predictive power between the simulated and online dataset, the online trained model was saved and reloaded to predict the simulated dataset, and it was observed that the online trained model achieved an accuracy of 92.47%, as in Table 7. This may justify the validity of the simulated data for another research purpose in the future.

Table 7: Simulated data prediction with the ANN model for online data

Classification Metric	Score Value (%)
Precision	91.2972
Recall	88.1141
Accuracy	92.4706

4.1.1 TensorFlow Lite Model Conversion

TensorFlow library was first imported into the model development environment for the conversion of the saved ANN model into a TensorFlow compatible model, as shown in Figure 13. The compatible model was saved with ***tf_model.h5.***

```
In [21]:
#tensorflow
import tensorflow as tf
#Save model with tensorflow
tf.keras.models.save_model(
    loaded_model,
    "tf_model.h5",
    overwrite=True,
    include_optimizer=True,
    save_format=None
)
```

Figure 11: Conversion of the saved model into TensorFlow compatible model

The TensorFlow compatible model was converted to an android mobile device compatible using the ***TFLiteConverter,*** as seen in Figure 14.

```
In [22]:
import warnings
warnings.filterwarnings('ignore')


tfloaded_model = tf.keras.models.load_model('tf_model.h5')
converter = tf.lite.TFLiteConverter.from_keras_model(model=tfloaded_model)

tfmodel = converter.convert()

# Save TFLite model into a .tflite file

open("./Data/accident_model.tflite","wb").write(tfmodel)
```

Figure 12: Conversion to Android Device compatible model using TFLite Converter

The output was saved in the data folder with the file ***accident_model.tflite.*** The saved accident model was imported into Android Studio in Figure 15. Android Studio makes the process easier with a few clicks.

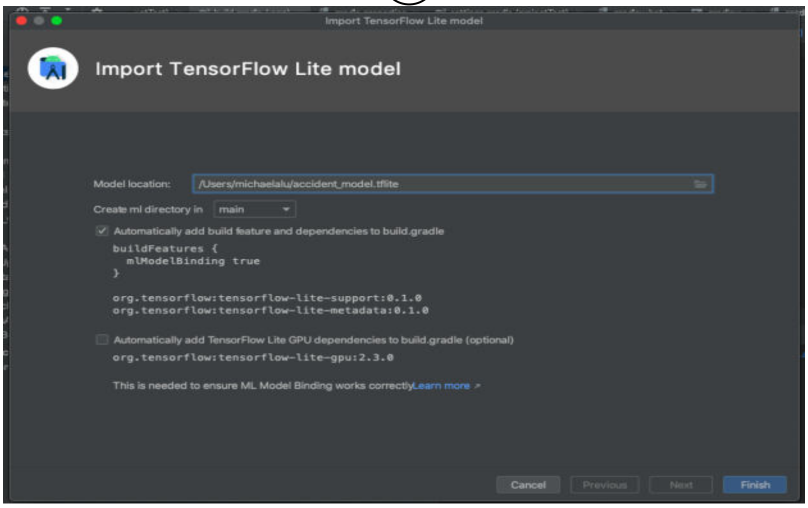


Figure 13: Selecting the saved accident model TFLite file

An instance of the imported model class was called on the sensor event as described below in the code. The new instance of the model was used to process the converted float array representing the generated sensor data. The input dimension for detection is {1,5}, which represents the *accelerometer_x, accelerometer_y, accelerometer_z, gravitional_force_magnitude, and sound amplitude*. The input object was processed, and the output object was generated for testing. The whole operation is asynchronous; therefore, detection can be done in real-time. In the case of error, the error exception was caught to avoid crashing the application.

4.2 System User Interface

4.2.1 Authentication Interface

According to our use case diagram, the system developed involved the major entity, which is the normal user (victim) and the emergency authority or office. The system requires both entities to register on the system to use the system. Figure 16 & Figure 17 describes the registration and login activity, respectively. In this activity, users' information is collected for further usage in the system.

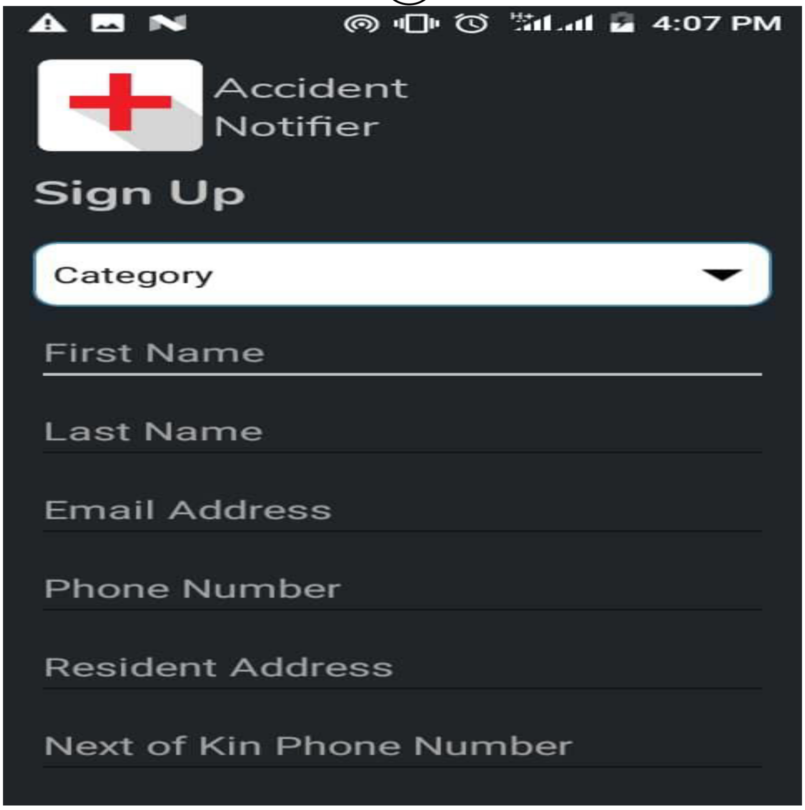


Figure 14: Registration Activity

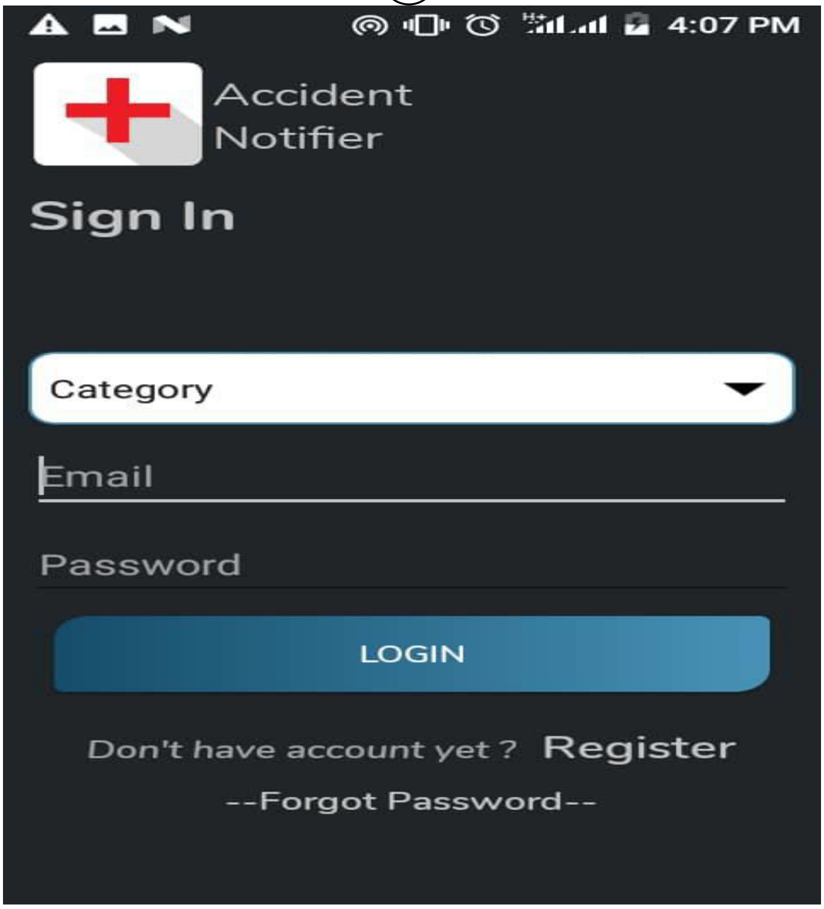


Figure 15: Login Activity

4.2.2 Accident Detection and Reporting Interface

After a victim is authenticated, the next activity screen is the home activity. The authentication is once, which means, in a subsequent launch, the application opens on the home activity. The detection and notification of an accident handle the main functionality of the system. Several widgets can be seen on the home activity, as illustrated in Figure 18, including the countdown timer widget, the manual notification button, the stop notification button, and so on. On detection of an accident event, the notification delay countdown starts; in this study, 2 minutes delay time will be considered. Users can

ISSN PRINT 2811-3187 ONLINE 2811-3209 Volume 3 NO 3 2024 Conf. Edition

manually initiate a notification or stop the delay timer, as seen in Figure 19 (Emergency Authority home activity interface). The interface majorly displays the push notifications from the firebase real-time database as in Figure 20, i.e., the information of any user involved in the accident.

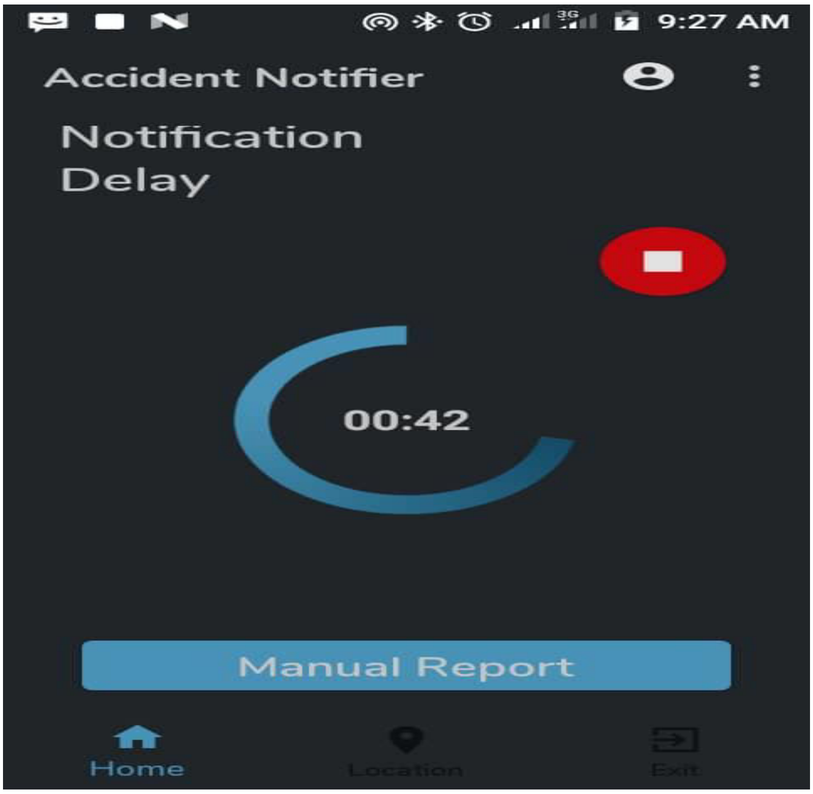


Figure 16: Accident Detection (Victim)

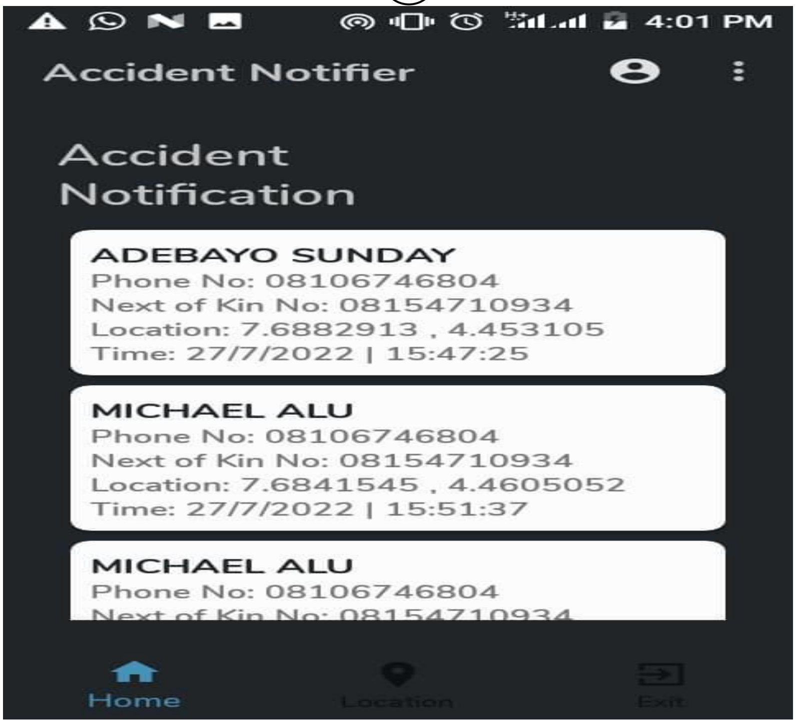
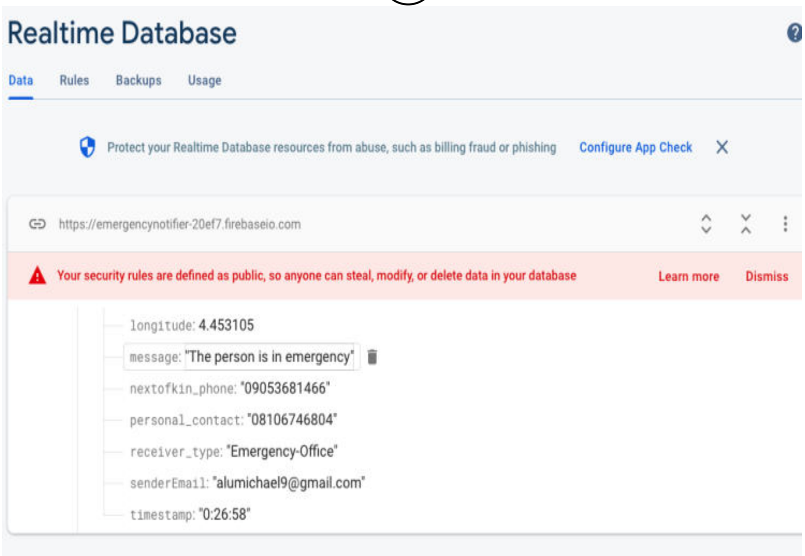


Figure 17: Accident Report (Authority)

Figure 20 shows an instance of any generated report sent to the emergency authority on detecting an accident event. According to the firebase database schema, the location coordinate and other user information were sent to the emergency authority and the contact person.


Realtime Database
Data
Rules
Backups
Usage
Protect your Realtime Database resources from abuse, such as billing fraud or phishing
Configure App Check
https://emergencynotifier-20ef7.firebaseio.com
Your security rules are defined as public, so anyone can steal, modify, or delete data in your database
Learn more
Dismiss
longitude: 4.453105
message: "The person is in emergency"
nextofkin_phone: "09053681466"
personal_contact: "08106746804"
receiver_type: "Emergency-Office"
senderEmail: "alumichael9@gmail.com"
timestamp: "0:26:58"

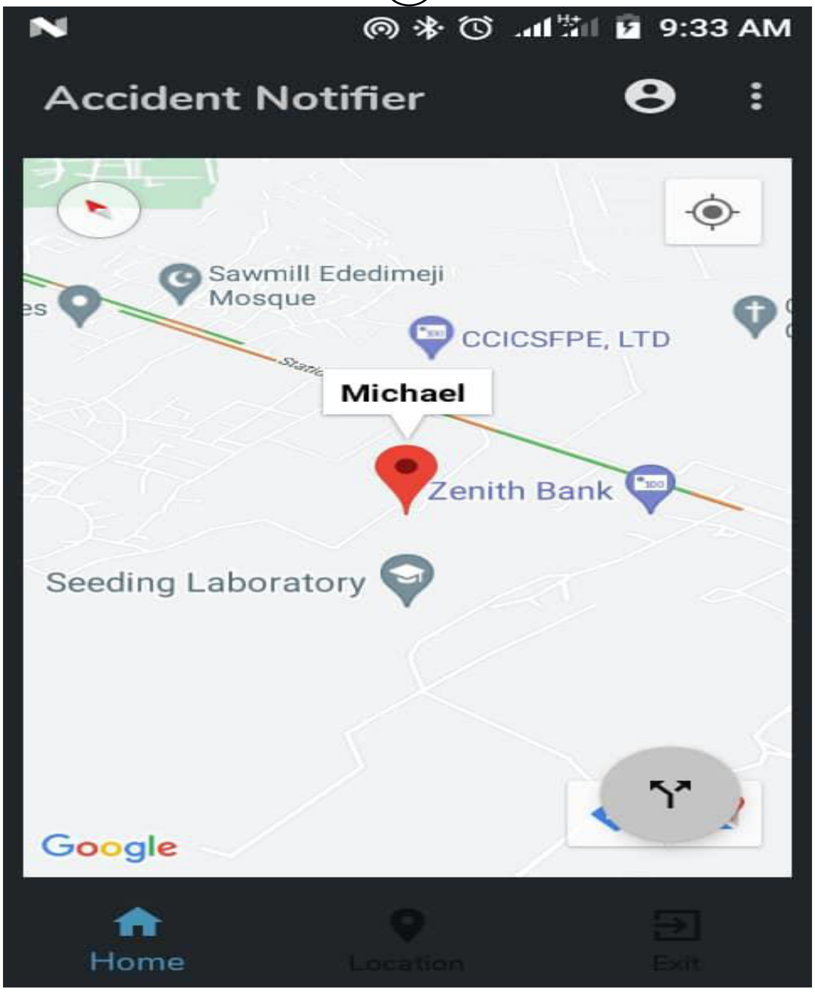


Figure 18: Map Activity for user tracking

4.3 System Evaluation

One of the difficulties in carrying out accident detection exercises is the cost of real-life experimentation or evaluation. Therefore, simulation is the best, fastest, and most inexpensive

method for accident detection. Based on the set of features used in the detection of accident event in this work, the system is evaluated by considering several test cases as summarized in Table 8.

Table 8: Summary of system test cases

Test Input						Output
accelerometer x-axis		accelerometer y-axis	accelerometer z-axis	audio	G-Force	
Case 1	0.996	0.641	9.174	0	0	No Accident Detected
Case 2	2.106	2.106	7.34	0	1.2347	No Accident Detected
Case 3	0	0	0	105	0	Accident Detected
Case 4	2.106	0.919	7.23	101	0.7733	Accident Detected
	0.574	-0.83	0.548	30	9.787	No Accident

Case 1: The system was tested with an accident instance of simulated data so that other features (G-force and audio data) were set to 0, except for accelerometer data. The result shows NO ACCIDENT was detected in this case.

Case 2: The system was tested with only an accelerometer and gravitation force magnitude instance of simulated data while setting the audio data to 0 and evaluating the log result. The test result shows NO ACCIDENT DETECTED as the audio data was set to 0 in this case, even though other features were correlated to the accident sample.

Case 3: The system was tested with only audio data while setting the other features to 0 and evaluating the log result. The test was conducted to check if the system may trigger an accident even when the accelerometer or gravitational force magnitude is 0, that is, on an undisturbed accelerometer or gravitation due to gravity state. It turned out that the system classified this case as ACCIDENT with a very high sound amplitude. The point is that the system may still detect accidents accurately even when the mobile phone is intact without disturbance. According to this test case, an audio amplitude of about 105 decibels is considered dangerous and may possibly be an accident event, although this may not be every case.

Case 4: The system was randomly tested with the "accident" and "no accident" samples while the result was compared.

Testing with a positive accident sample [2.106, 0.919, 7.23, 101, 0.773328174, 1], where the first five elements represent an instance of the input features, and the last element means the output 1 is ACCIDENT.

Testing with "no accident" sample [0.574, -0.833, 9.548, 30, 0.978740162, 0], where the last element represents a negative label 0, that is, no accident occurred. The sample was tested with the mobile application to test the system's effectiveness, and as shown in Table 8, the result turned out as expected, NO ACCIDENT.

4.4 Comparison of Result

This study has developed a real-time mobile-based accident detection and notification system with the simple and available resources on every mobile

device: the modal data. This study stepped further by introducing these available modal data (accelerometer, gravitational force, and audio data) into deep learning model (ANN(MLP)).

At the end of the experiment, it was observed that the ANN model achieved an accuracy of 99.5% on the simulated data and 99.8% on the in-vehicle monitoring data from Honda pilot vehicles captured on May 4th, 7th, and 8th of 2018. The ANN model was used to develop the mobile application system since it format (.H5) was supported by the TensorFlow-lite framework. developed mobile application performed as expected. In that, the cross-validated ANN model integrated into the current study achieved a better result compared to the existing research [9] [17]

The system development ensures everyone has access to accident detection and automated notification, not that alone. The test cases run on the the summary of the result comparison is given in Table 9, which shows that the current study performed well compared to the existing accident detection systems.

Table 9: Result Comparison with Related Studies

S/N	Author (Year)	Input Data Type	Result
1	Lu et al. (2020)	Crash Videos	Research achieved an accuracy of 87.78% on the testing dataset and an acceptable detection speed (FPS > 30 with GTX 1060).
2	Yang et al. (2020)	Crash Images	The AUC of the CNN-XGBoosting event detection method was found to be greater than 99%, which is significantly better than GBoost, CNN, SVM, and GBDT.
3	Shravan et al. (2021)	Sensor Data	The system increases the accuracy compared to previous work with a single sensor.
4	Sebastian et al. (2021)	Sensor Data	Although the research uses a hardware approach for detecting an accident, the results of the experiments show that the proposed system performs as intended.

5	**Current Study**	Data World Dataset (165Kb), Simulated Accident Tabular Dataset (57Kb) Characterized by an accelerometer, gravitational force, and sound.	In this work, we have developed a mobile-based accident detection and notification with 99.5% detection accuracy with ANN model. All system evaluation test cases passed. Internet service is optional in sending notifications.

Conclusion and Future Work

One of the factors responsible for fatalities and lifelong disabilities globally is road accidents. The predominant increase in mortality rate on the highway has been attributed to the late arrival of emergency authorities, which could be because of delays in reporting time. Some of the accident events were not even communicated. This research has developed a mobile- based accident detection and notification system with multi-modal data. The detection system has incorporated trained simulated data that is validated by unseen data online. The simulated data was characterized by some selected modal data equivalent to the online features. The simulated data was trained using the multilayered perceptron artificial neural network (ANN) model. The trained model was tested using online data from the data world platform. At the end of the experimentation, it was observed that the ANN model achieved an accuracy of 99.5% on the simulated data and 99.8% on the online dataset. The converted model was imported into the mobile application for real-time accident detection, and several test cases were experimented with to validate the system's accuracy. At the end of the system test cases, it was observed that the system worked as expected, passing all test cases. The system also introduced SMS and Firebase push notification service for the notification of accident events—Geolocational API for registered emergency authority to track the accident location.

This work has played a vibrant role in providing the general public with the accident detection system, eliminating the restriction to usable geographical location. In addition, the system is relatively inexpensive since there is no need for the integration of intelligence transportation system into vehicles.

References

[1]C. O. Ojabo, O. S. Adeniyi, and B. A. Ojo, "An Assessment of Visual Acuity of Commercial Vehicle Drivers in Makurdi, Benue State, North-Central Nigeria," *Current Journal of Applied Science and Technology*, vol. 39, no. 18, pp. 61–67, 2020,doi: 10.9734/cjast/2020/v39i1830772.

[2]R. Ramkumar, S. Dinesh, S. N. Kumar, and G. Prathipa, "Smart Alert System for Vehicles," vol. 1, no. 5, pp. 32–39, 2016.

[3]K. Palanichamy, "Chapter 19 - Integrative Omic Analysis of Neuroblastoma," in *Computational Epigenetics and Diseases*, vol. 9, L. K. Wei, Ed. Academic Press, 2019, pp. 311–326. doi: https://doi.org/10.1016/B978-0-12-814513-5.00019-2.

[4]R. K. Mishra, G. Y. S. Reddy, and H. Pathak, "The Understanding of Deep Learning: A Comprehensive Review," *Mathematical Problems in Engineering*, vol. 2021, 2021, doi: 10.1155/2021/5548884.

[5]B. Kumeda, Z. Fengli, A. Oluwasanmi, F. Owusu, M. Assefa, and T. Amenu, "Vehicle Accident and Traffic Classification Using Deep Convolutional Neural Networks," *2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP 2019*, no. June, pp. 323–328, 2019, doi: 10.1109/ICCWAMTIP47768.2019.9067530.

[6]A. Prasad, A. K. Tyagi, M. M. Althobaiti, A. Almulihi, R. F. Mansour, and A. M. Mahmoud, "Human activity recognition using cell phone-based accelerometer and convolutional neural network," *Applied Sciences (Switzerland)*, vol. 11, no. 24, 2021, doi: 10.3390/app112412099.

Fanca, A. Puscasiu, H. Valean, and S. Folea, "A Survey on Smartphone-Based Accident Reporting and Guidance Systems," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 9, pp. 409–414, 2018, doi: 10.14569/IJACSA.2018.090952.

[8]M. A. Anwer, S. M. Shareef, and A. M. Ali, "Optimized frame detection technique in vehicle accident using deep learning," *Zanco Journal of Pure and Applied Sciences*, vol. 32, no. 4, 2020, doi: 10.21271/zjpas.32.4.5.

[9]T. Sebastian and A. Emerson, "Efficient Accident Detection a nd Notification System," vol. 10, no. 07, pp. 334–337, 2021.

[10]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, doi: 10.14569/IJACSA.2018.090648.

[11]Nancy. P. D. Rao. D. Babuaravind. G. Bhanushree.S, "Highway Accident Detection and Notification Usingmachine Learning," *International Journal of Computer Science and Mobile Computing*, vol. 9, no. 3, pp. 168 – 176, 2020, [Online]. Available: https://ijcsmc.com/docs/papers/March2020/V9I3202026.pdf

[12]S. Mothe, A. S. Teja, B. Kakumanu, and R. K. Tata, "A model for assessing the nature of car crashes using convolutional neural networks," *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 3, pp. 859–863, 2020, doi: 10.30534/ijeter/2020/41832020.

T. Kalyani, S. Monika, B. Naresh, and M. Vucha, "Accident detection and alert system," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 4S2, pp. 227–229, 2019.

[14]J. G. Choi, C. W. Kong, G. Kim, and S. Lim, "Car crash detection using ensemble deep learning and multimodal data from dashboard cameras," *Expert Systems with Applications*, vol. 183, p. 115400, 2021, doi: 10.1016/j.eswa.2021.115400.

[15]R. Shah, H. Shaikh, and D. Shetty, "Survey of accident detection systems," vol. 1, no. 4, pp. 1–8, 2021.

Z. Lu, W. Zhou, S. Zhang, and C. Wang, "A New Video-Based Crash Detection Method: Balancing Speed and Accuracy Using a Feature Fusion Deep Learning Framework," *Journal of Advanced Transportation*, vol. 2020, 2020, doi: 10.1155/2020/8848874.

[17]B. R. Shravan, K. Sudarshan, and V. K. Bhat, "Accident detection and reporting system," *International Research Journal of Engineering and Technology*, vol. 8, no. 7, pp. 3038–3042, 2021.