J-Component Project Report

Course Name: Internet of Things

Course Code: CSE3009

Slot: B2

Semester: Winter Semester 2020-21

TITLE:

POTHOLE DETECTION USING IOT PRINCIPLES

Submitted by:

Devina Varshney 18BCE0741
Rishabh Kumar 18BCE0816
Ankit Mishra 18BCE0780

1. INTRODUCTION

Today's autonomous driving applications are relying on autonomous vehicle systems in which the needed information is gathered, processed and analysed on the vehicles themselves. In this way the Automated vehicle systems consist of inputs from a large variety of sensors, data signal condition and decision making by central or edge processing units and outputs to a large variety of actuators. The overall objective is to bring together relevant knowledge and technology from the automotive and the IoT value chains in order to develop IoT-architectures and platforms which will bring Automated Driving towards a new dimension.

Smart sensors and actuators in the vehicles, roads and traffic control infrastructures collect a variety of information to serve enhanced automated driving, while considering the timing, safety and security constraints. Road surface anomalies, such as potholes, speed bumps, railroad crossing and joints, can determine some problems for vehicles and can affect road users' safety. Road quality assessment plays a key role in infrastructure management and it is useful to an adequate allocation of road maintenance operations.

Road surface anomalies, such as potholes, speed bumps, railroad crossing and joints, can determine some problems for vehicles and can affect road users' safety. Road quality assessment plays a key role in infrastructure management and it is useful for an adequate allocation of road maintenance operations. A pothole refers to a shallow pit on a road's surface, caused by activities like erosion, weather, traffic and some other factors.

It becomes essential to monitor road and surface quality to make sure that there are no unforeseen circumstances. Owing to the gravitation of human needs towards a smart city, and owing to decrepit road infrastructure our aim is to constantly monitor the road surface to improve the quality and ensure that the car journey is safe. Detecting and hence avoiding potholes may reduce the fuel consumption, wear-tear and maintenance cost of a vehicle. Autonomous vehicles can make the right manoeuvres to avoid potholes or other dangerous situations. It is desirable to have a mechanism for detecting the condition of roads and get them repaired as soon as possible. As a result, working on monitoring road conditions has gained significant attention in recent times.

To help detect the potholes on the road, we will be using IoT principles and monitor the quality of roads. We plan to build this with the help of an accelerometer, further we also plan to store the data on a cloud platform.

Till now, smartphones used for road condition monitoring, are limited to recording of accelerations, processing them to discern potholes and monitoring the overall condition of road surfaces. Therefore, the data must be pre-processed before it can be used. This can be done for example by using a passband filter. It removes low and high frequencies from the measured data. This makes the data much cleaner and easier to process. Data can also be divided into small segments and normalized to some specific scale to make the feature extraction and classification easier

Further we will retrieve this data to perform analysis using different algorithms and comparing their performances. The algorithm will be used to initially train on a dataset and then using a particular threshold value we will check whether there is a pothole and this would instantly trigger a notification.

1.1 AIM

Many times, when we travel on the road, especially through rain-affected cities, it becomes a constant need to monitor road quality to essentially make sure that there are no unforeseen circumstances. Owing to the gravitation of human needs towards a smart city, and owing to the decrepit road infrastructure, our aim is to constantly monitor the road surface to improve the quality and ensure that the car journey is safe. We aim to build a mobile app, which collects data from in-built sensors of the phone. The collected data is sent to a database built using Google's cloud platform, using a socket connection. The socket connection enables the data to be sent Realtime. Google's Firebase Database offers a Realtime database, which can be deemed appropriate for our purpose. The analysis involves fetching of the stored data, the fetched data is further cleaned & converted into a suitable form. The Z-Thresh algorithm considers the minimum value of the z-axis accelerometer as a threshold. Analysis helped us accurately determine the location of the potholes.

1.2 OBJECTIVE

Road surface conditions are major distractions for safe and comfortable transportation. To ensure the safety we need to first identify the flaws on roads like potholes. There has been some work done in this field which had some constraints like memory constraints as a lot of data is collected during the process. Some algorithms previously applied have some performance limitations and there has been a lack of precision and accuracy. To counter this, we will use first find out the pothole through data collected from the app (which is used as an accelerometer) then that data will be sent to the cloud for analysis and using z-threshold algorithm we will accurately predict that if there is a pothole or not if it is there, the location of all the pothole will be sent to the authority via SMS which will inform them about it so that they can fix it.

1.3 BENEFITS

- Safe travel on road.
- Informing authority about the pothole so that it can be repaired as soon as possible.
- Instant alert of a pothole
- Website to track the locations of pothole, accessible to the authorities
- Socket connection ensure reliable & quick transmission of data.

A literature survey was done, to understand about the existing work done in the domain of pothole detection. This survey evaluated the methodology, algorithms and other parameters of the study to help compare and contrast each of the existing methods.

The literature survey has compared the pros & cons of each of the evaluated papers. The literature survey has is given in the following section.

2. LITERATURE SURVEY

TITLE	ADVANTAGES	DISADVANTAGES/ GAPS
A three-tier road condition classification system using a spiking neural network model	Using a Spiking Neural Network (SNN) yields significantly higher classification performance when compared to a Support Vector Machine (SVM) and Multilayer Perceptron (MLP) trained and tested using the collected datasets and classification models.	The system has memory constraints as a lot of memory is consumed in the process, which makes the system slow.
Vehicle Vibration Signal Processing for Road Surface Monitoring	The system can detect and classify the abnormal events such as pothole and hump from the collected data at any vehicle driving speed. It can also estimate the severity of the identified event when the test-driving speed falls in the range of 15 to 20 km/h.	The system is unable to detect the irregularities for only a particular range and is unable to estimate the event severity at any given vehicle driving speed.
Evaluation of Detection Approaches for Road Anomalies Based on Accelerometer Readings—Addressing Who's Who	Use the smartphone as a cheap and widespread tool to measure the quality of roads. Evaluating a set of seven seminal heuristics that have guided and influenced the development of new anomaly detection strategies. The best strategy, among the seminal works, is called STDEV(Z), which clearly outperforms other popular and widely known detectors such as Pothole Patrol and Nericell. Measures of dispersion, specifically standard deviation, are among the best indicators to identify disruptions on accelerometer readings.	Certain circumstances where STDEV(Z) and SVM(Z) showed their lowest performance.
Road pavement condition diagnostics using smartphone-based data crowdsourcing in smart cities	The paper uses the road condition tool (RCT) to analyze data (linear accelerations, speed, and vehicle location) from a large number of RCT-enabled smartphones. The combined data can give us an accurate idea about the road conditions of a pavement. An accuracy of over 90% with a precision of 80% was obtained.	Accuracy & Precision while are very impressive in terms of machine learning algorithms, these numbers are very hard to work with in the real world. A 10% error would mean thousands of 'false-positive' detections, which would lead to a range of problems.

Scalable Cloud–Sensor Architecture for the Internet of Things	The paper uses CEB (Cloud-Edge-Beneath), an architecture for large-scale, extensible cloud–sensor systems, along with an event-based programming model for IoT application development. They conducted experiments to validate CEB's scalability and capability of adapting to dynamic load on the cloud–sensor system.	Even though CEB is an architecture designed with a built-in optimization platform, this paper does not propose or report on any optimizations to enable further scalability and energy-efficiency of the cloud–sensor system

Table 1: Literature Survey

3. DRAWBACKS OF THE EXISTING SYSTEMS

3.1 Speed:

Speed: Detecting irregularities of the road surface and detection of potholes and humps is only applicable for a certain speed range and is not possible for any given speed which falls outside of this range.

3.2 Accuracy:

Accuracy & Precision while are very impressive in terms of machine learning algorithms, these numbers are very hard to work with in the real world. A 10% error would mean thousands of 'false-positive' detections, which would lead to a range of problems.

3.3 Optimizations:

While Scalable IOT-Cloud Infrastructures do exist, the networks are highly unoptimized, which not only leads to the structures being unnecessary complicated, but also higher energy consumption overall.

The next section explains about the sensors involved while detecting a pothole. Each of the sensor's working has been explained in-depth as well.

4. COMPONENTS USED

The accelerometer & GPS sensors of the smartphone are used to fetch the acceleration along all the axes & the position of the user in terms of latitude & longitude. This section explains the working of each of the sensors involved.

4.1 Sensors

Smartphone Accelerometer to find the value of acceleration of the x,y,z axis and Global Positioning System module for finding the latitude and the longitude have been used as sensors.

4.2. Working model of sensors

4.2.1 Accelerometer

The basic underlying working principle of an accelerometer as shown in Figure 2 is such as a dumped mass on a spring. When acceleration is experienced by this device, the mass gets displaced till the spring can easily move the mass, with the same rate equal to the acceleration it senses. Then this displacement value is used to measure the acceleration.

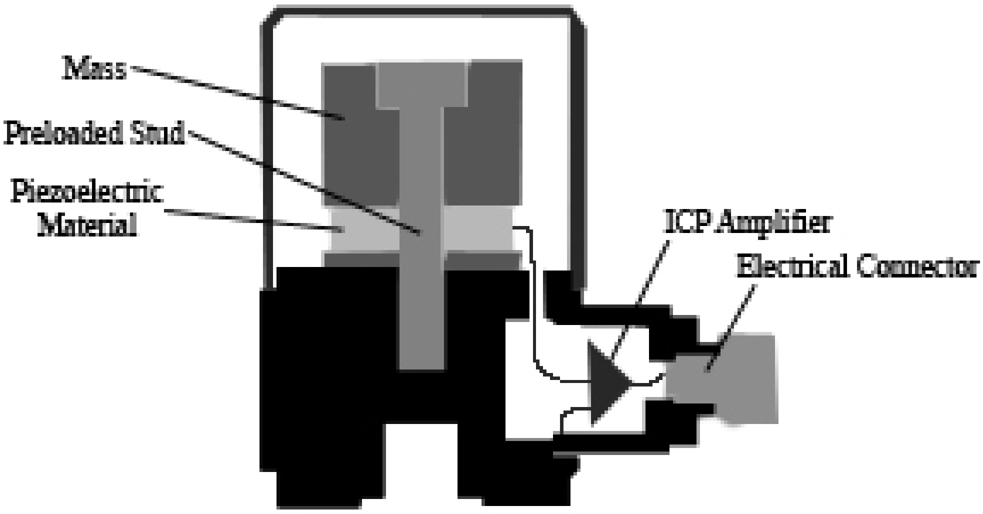

Fig 2: Accelerometer

In digital devices like mobile phones accelerometers are designed using different methods. Piezoelectric, piezoresistive and capacitive components are generally used to convert the mechanical motion caused in an accelerometer into an electrical signal. Piezoelectric accelerometers are made up of single crystals. These use the piezoelectric effect to measure the acceleration. When applied to stress, these crystals generate a voltage which is interpreted to determine the velocity and orientation.

Capacitive accelerometers use a silicon micro-machined element. Here capacitance is generated when acceleration is sensed and this capacitance is translated into a voltage to measure the velocity values. Modern accelerometers are the smallest MEMS, consisting of a cantilever beam with proof mass. Accelerometers are available as two-dimensional and three-dimensional forms to measure velocity along with orientation. When the upper-frequency range, high-temperature range, and low packaged weight are required, piezoelectric accelerometers are the best choice. The accelerometer is used to sense orientation, coordinate acceleration, vibration, and shock through a smartphone to gather data and analyse that to find out if there is a pothole or not when a vehicle is running on the road.

4.2.2 Global Positioning System

Global Positioning System (GPS) which is a satellite navigation system that furnishes location and time information in all climate conditions to the user. GPS is used for navigation in planes, ships, cars, and trucks also. The system gives critical abilities to military and civilian users

around the globe. GPS provides continuous real-time, 3-dimensional positioning, navigation, and timing worldwide. GPS is a radio navigation system. It uses radio waves between satellites and a receiver inside your phone to provide location and time information to any software that needs to use it. You don't have to send any actual data back into space for GPS to work; you only need to be able to receive data from four or more of the 28 satellites in orbit that are dedicated for geolocation use.
Each satellite has its own internal atomic clock and sends a time-coded signal on a specific frequency. Your receiver chip determines which satellites are visible and unobstructed (that's important, and you'll read why in a bit) then starts gathering data from the satellites with the strongest signals. GPS data is slow, and this is by design — satellites run on rechargeable batteries, and sending a fast signal hundreds of thousands of miles would require more power-so it'll take up to a minute to get your geolocation.
The user's phone's GPS receiver uses the data from these signals to triangulate where you are and what time it is. Notice the word triangulation and the mention above those four satellites are required for GPS to work. The fourth signal is used to determine altitude so you can get your geolocation data on a map with only three signals. Assisted Global Positioning System (AGPS) adds cellular location data to assist geolocation. Your phone carrier knows where you are since your phone "pings" cell towers. How precise this is will depend on the strength of the signal between your phone and the tower, but it's usually good enough to be used for location data.

5. PROPOSED ARCHITECTURE

The data collected from the sensors of the phone are sent to the Web server via a Socket Connection. The socket connection enables the data to be sent much quicker as compared to the traditional HTTP Protocol. The data sent is then fetched, to be analysed. The analysis is done using the Z-thresh algorithm, which helps determine the location of the pothole. The location of the potholes is further sent to the authorities.

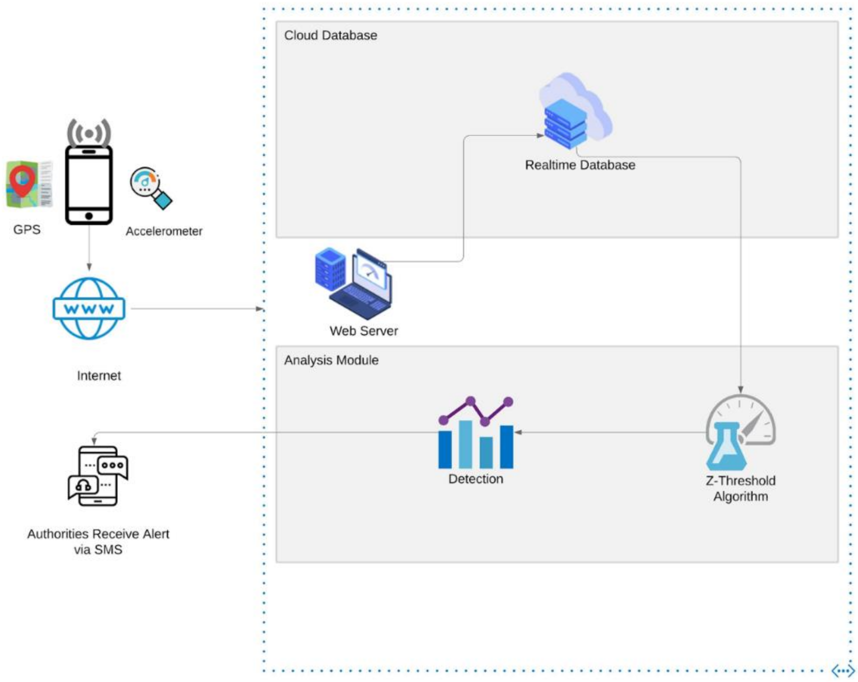

Fig 1: Proposed system model

5.1 WORKING PRINCIPLE

5.1 Z-Thresh Algorithm

Z-Thresh is the simplest event detection algorithm. To detect potholes the Z-Thresh algorithm considers the minimum value of the z-axis accelerometer as threshold. In the proposed model we have used a threshold of 0.4*g for classification. The algorithm calculates the value of the received data with that of threshold value. If the amplitude of the value exceeds the threshold then a pothole is identified. Figure 4 shows the Z-thresh algorithm where the events exceed the given threshold value. If the magnitude of value of z-axis deflects firstly in positive direction and then towards negative direction with slight deflection, then it will indicate a pothole.

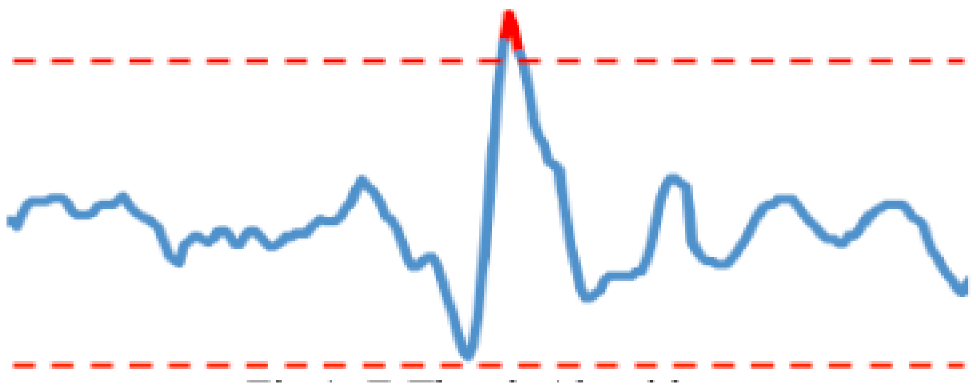

Fig 3: Z-Thresh Algorithm

The pothole will be indicated with the change in the value of z-axis: in this case an alert is sent to the Firebase every time a pothole or possible bump is detected.

In the proposed model, Google's Cloud Platform has been used to store the data, which is collected from the Mobile App. The Data is later fetched from Cloud, and analysed further.

Realtime Database is a cloud-hosted NoSQL database with SDK support for iOS, Android, and the web. It easily integrates with Firebase's other tools for authentication, file storage, analytics and others. This tool stores data in JSON documents, so everything is either a key or a value. Data synchronization uses web sockets, allowing for very snappy transactions.

A WebSocket is a persistent connection between a client and server. WebSocket's provide a bidirectional, full-duplex communications channel that operates over HTTP through a single TCP/IP socket connection. At its core, the WebSocket protocol facilitates message passing between a client and server. The idea of WebSocket's was borne out of the limitations of HTTP-based technology. With HTTP, a client requests a resource, and the server responds with the requested data. HTTP is a strictly unidirectional protocol — any data sent from the server to the client must be first requested by the client. Long-polling has traditionally acted as a workaround for this limitation. With long-polling, a client makes an HTTP request with a long timeout period, and the server uses that long timeout to push data to the client. Long-polling works, but comes with a drawback — resources on the server are tied up throughout the length of the long-poll, even when no data is available to send.

WebSocket's, on the other hand, allow for sending message-based data, similar to UDP, but with the reliability of TCP. WebSocket uses HTTP as the initial transport mechanism, but keeps the TCP connection alive after the HTTP response is received so that it can be used for sending messages between client and server. WebSocket's allow us to build "real-time" applications without the use of long-polling.

WebSocket's begin life as a standard HTTP request and response. Within that request response chain, the client asks to open a WebSocket connection, and the server responds (if it is able to). If this initial handshake is successful, the client and server have agreed to use the existing TCP/IP connection that was established for the HTTP request as a WebSocket connection. Data can now flow over this connection using a basic framed message protocol. Once both parties acknowledge that the WebSocket connection should be closed, the TCP connection is torn down.

6. IMPLEMENTATION

Software on your phone feeds this raw cellular location data to the GPS receiver, which will periodically switch between GPS data and cellular location to get a very close approximation (within 50 meters or so) in real-time. In other words, GPS can use data collected by your phone from the cell site it is connected to in order to work faster and more accurately.
To use the accelerometer reading for detecting various events, it is possible to virtually reorient the axes of the smartphone to align along the axes of the vehicle. Readings from the reoriented axes can be used to detect events. Leveraging an accelerometer as a vibration sensor, the characterization of potholes and roads can be done using the readings of the accelerometer. It has been implemented in an App in Android Studio to collect the raw accelerometer data in the three directions and send the data to a socket.
The use of the mobile device based on mobile sensing techniques to detect potholes, is suitable and convenient: all the motion sensors return multidimensional arrays of sensor values for each SensorEvent. The linear acceleration sensor provides a three-dimensional vector representing acceleration along each device axis, excluding gravity. The simplest way to remove the offset of a linear acceleration sensor, is to build a calibration step into the application, in order to iterate the alignment of the smartphone accelerometer's coordinate system and the vehicle's coordinate system. Smartphone accelerometers use the standard sensor coordinate system. Some sample accelerometer readings and GPS readings have been given in Figure 3.
In practice, this means that the following conditions apply when a device is lying flat on a table in its natural orientation:

- If the device is pushed on the left side (so it moves to the right), AccX value is positive.
- If the device is pushed on the bottom (it moves away from you), AccY value is positive.
- If the device is pushed toward the sky with an acceleration of A m/s^2, AccZ is equal to A+9.81, which corresponds to the acceleration of the device (+A m/s^2) minus the force of gravity (- 9.81 m/s^2).

The stationary device will have an acceleration value of +9.81, which corresponds to the acceleration of the device (0 m/s^2 minus the force of gravity, which is -9.81 m/s^2).

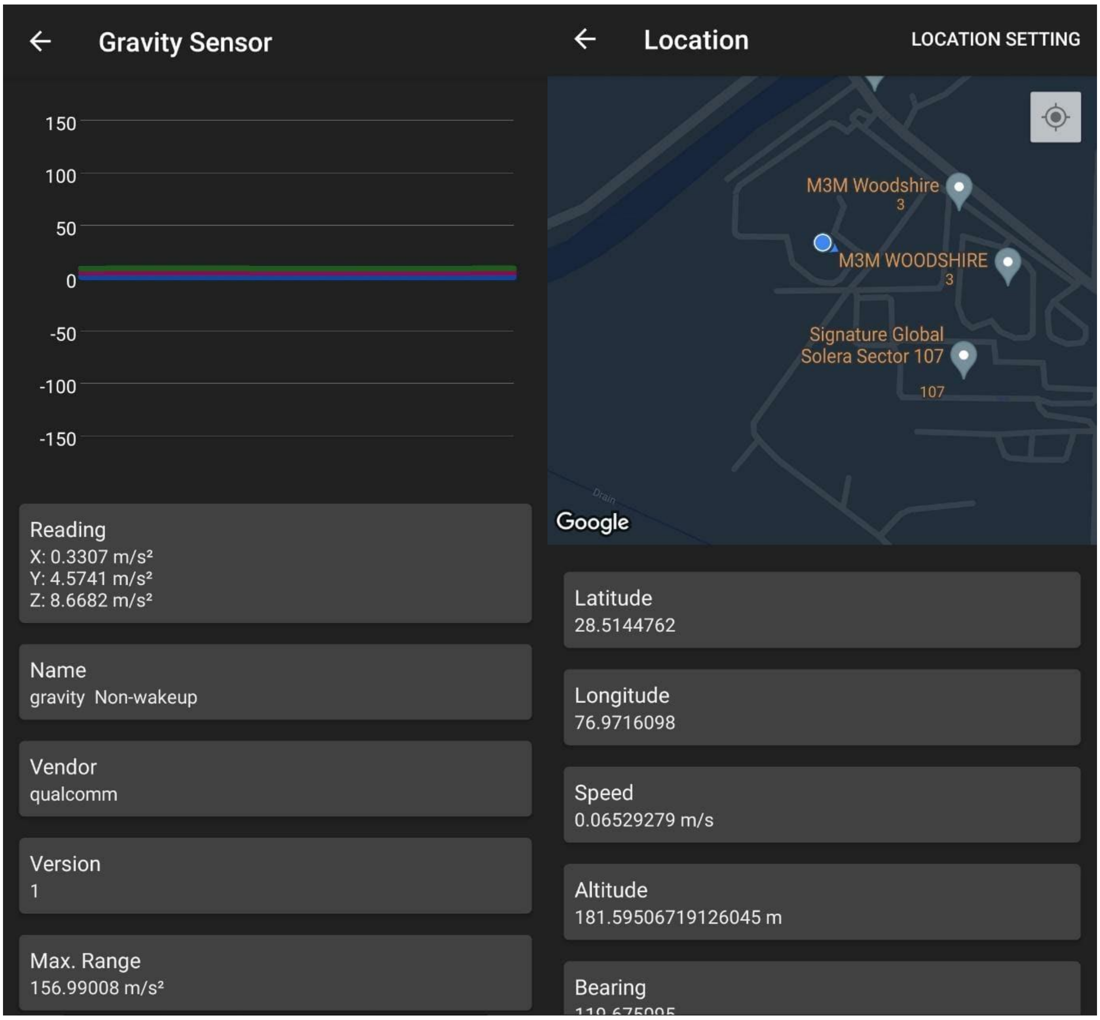

Fig 4: Sample accelerometer and GPS Readings

The accelerometer data is normalized and is adopted in the pothole detection algorithm to obtain the pothole information. The analysis reveals if a pothole is present at that particular location. The proposed Realtime pothole detection method based on mobile sensing includes three main steps:

- accelerometer data normalization,
- pothole detection algorithms, and
- pothole location determination.

The most common approach followed in previous studies for detecting road conditions is using sensors to recognize the vibration patterns of the vehicle caused due to any deformity or obstacle on the road. The proposed Realtime pothole detection method based on "virtual sensor" includes three main steps: accelerometer data acquisition and normalization, pothole detection approaches with algorithm of signal processing, and pothole location determination (GPS data).In order to implement a real-time algorithm, it is important to integrate the raw accelerometer signals with the GPS data periodically acquired by a database: in this case it is possible to provide information about localization of the holes. The raw accelerometer data is input to a processing algorithm that uses a Z-thresholding algorithm to signal the pothole.

6.1 WHAT FORM OF DATA ANALYTICS CAN BE APPLIED TO COLLECTED DATA

The first step for performing data analytics is data collection. Here the most essential step is the flow of data and tracking of this data. For that it is of absolute significance that a suitable time parameter is specified in order for the efficient monitoring of the data. For every n second, we can capture the data and then we can use that for analysis.

Firebase stores the values of the observations along with their latitude, longitude, x, y, z coordinate of the android phone. We first retrieve data from firebase by downloading the json file of the data available in firebase.

Once the file is available in json format we load the file in our script and convert it to csv format. Then the csv file is loaded for analysis. The next step is to filter the data that is the value of z with respect to indices. We set a certain threshold for z then find out the latitude, longitude, indices along with z value of the indices that cross this threshold.

Consider a situation that happens when there is an immediate jerk or a sudden turn. Such a situation will act as an outlier to this because the period is really short. For every observation there can be a comparison between the previous and next observation. If either of them is above the threshold implies we can choose them as an option. In other cases, we can treat the data observation as an outlier and not treat it as a pothole.

Based on this we are successfully able to collate a minimum of 2 such points.
Now our fundamental goal is to analyse the simultaneous occurrence of such potholes.

So now based on this we will find out the latitude and longitude of such an occurrence. Based on time and the Google Maps location when the event occurred, we can predict the location where the pothole had occurred.

This data can additionally benefit us to this extreme that if simultaneously n cars are going through this area, we can monitor the deviation from the threshold of all the cars and based on that we can make a weight by tyre analysis for the tyre companies as well. If certain areas have a situation where the same road gives a larger deviation from the threshold then it needs repair and suitably a note could be sent to the road managing authorities.

6.2. ALGORITHMS THAT CAN BE USED TO OVERCOME THE ISSUES

We are currently using the Z thresh algorithm for the same and the dataset is undergoing a binary classification. Our novelty stands at this precipice where we are eventually using grouped data or simultaneous occurring data and then pursuing the classification.

Additionally, other ideas could be using a KNN classifier. Based on the nearest neighbours we can classify the extent of the deviation from the normal threshold thereby highlighting the severity of the pothole and the road deformity. When we obtain such data, we can check based on Euclidean distance if the centres of the nearest neighbours are in the vicinity and then determine the extent of the accident.

Adjusting values for K is challenging at times because for small values, result may vary from those holding larger values of K and there might be cases when there would be a pothole whereas algorithms wouldn't predict it.

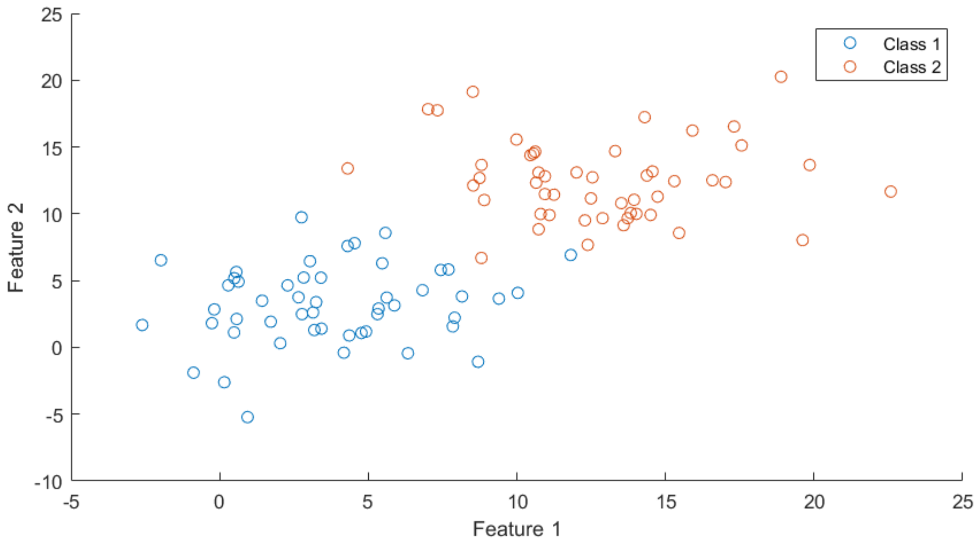

Fig 5: KNN Classification Algorithm

We could further use SVM (Support Vector Machine) to classify potholes on roads. In Support Vector Machine we can plot all x,y,z values in a 3-dimensional plane and draw hyperplane around the threshold values then determine whether they are close to the plane or away and correspondingly detect the pothole.
SVM algorithm is not suitable for large data sets. SVM does not perform very well when the data set has more noise i.e., target classes are overlapping. In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.
As the support vector classifier works by putting data points above and below the classifying hyperplane there is no probabilistic explanation for the classification.

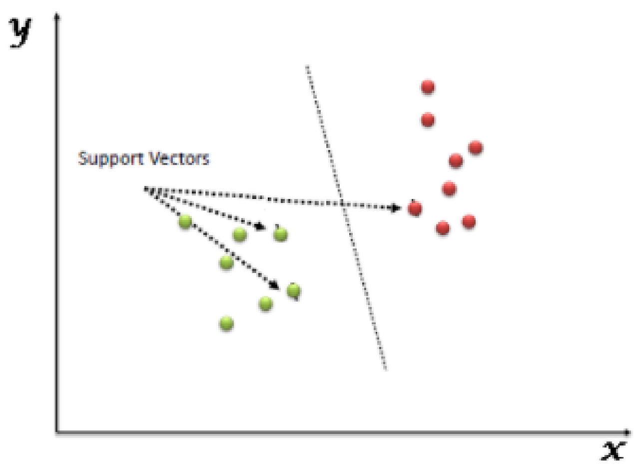

Fig 6: Support Vector Machine Algorithm

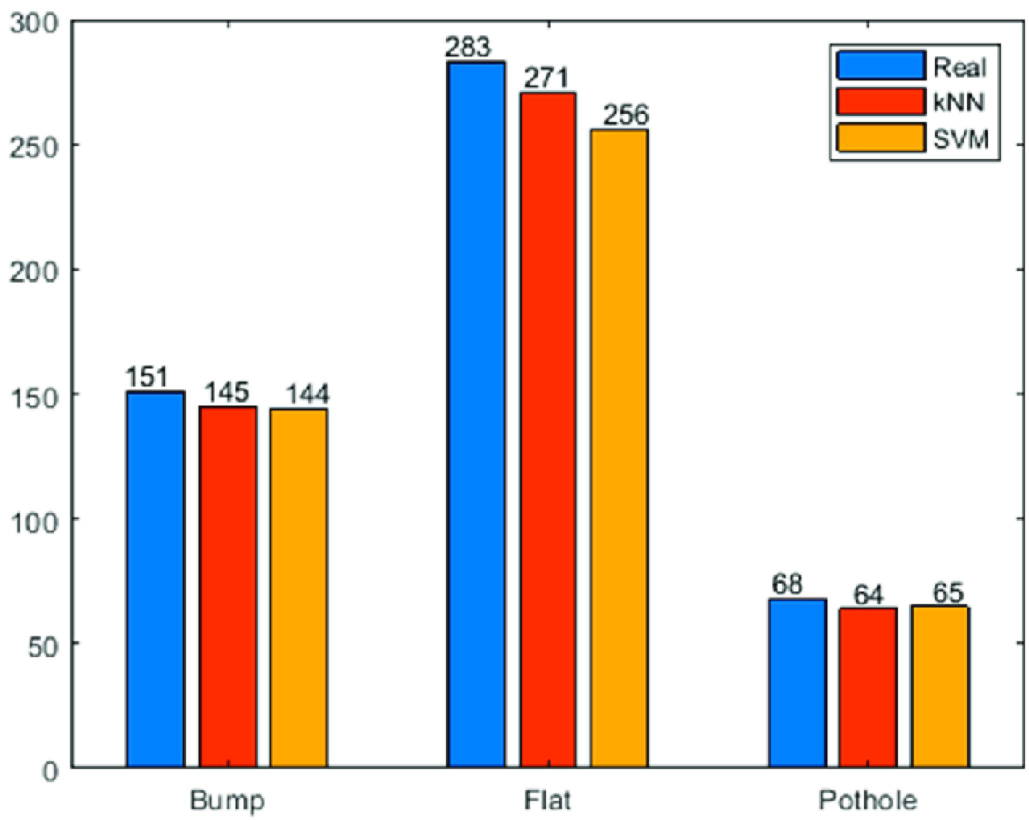

Fig 7: Performance Comparison of SVM and KNN with actual value

The other algorithm that can be used is Image classification where we use an image sensor to capture the images wherever a pothole appears. Then these images can be used in classifiers to come across more potholes and alert the road authorities on a frequent basis. Based on the road sensor.

Based on this we can classify the data into classes and keep snapshots of pothole filled roads and non-pothole-based roads.

Based on the 2 classes we can now run training models like VGG or Resnet 50 CNN model and train them to identify potholes or not. Now our image sensor that would capture a picture in a particular threshold of time would hence decipher the pothole based on our model and help us identify potholes.

Image processing however relies on colour and texture values of data. Additionally, the extent of the pothole cannot be mapped. Also, any new sudden data that it has not been trained on, might stimulate different results.

7. RESULTS AND DISCUSSIONS

The previous section discusses the working of the entire Pothole Detection system and also compares and contrasts it against some of the different approaches, involving concepts of machine learning. The results section would show the outcome of implementing the methodology proposed in the previous sections.

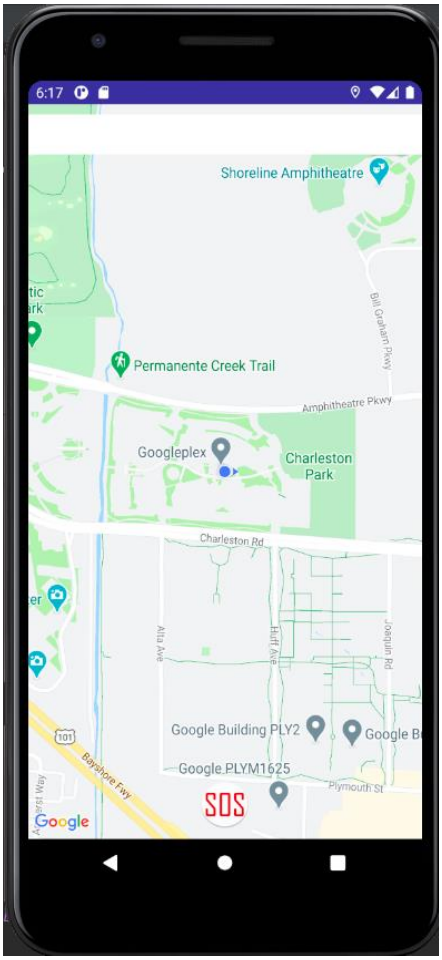

Fig 8: The display when user opens the application

Z-Threshold algorithm predicts a pothole based on the following conditions:

- The Value of Z-axis acceleration should be greater than 0.4*g
- The Value of the previous & the next observation should be lesser than the acceleration of the point where the pothole is present.

The algorithm detects whether both of the conditions have been satisfied. The algorithm also groups observations, which indicate the same pothole, i.e., observations which are right next to each other as shown in Figure 4.

7.1 Code Implementation

The piece of code given below is the one of the most crucial parts of the implementation of Z-thresholding algorithm. The array pothole_final contains the points that are suspected to be potholes. These points are found out by simply applying the condition of thresholding to all the z-axis acceleration values. The pothole_final array contains index of such points, thus, continuous indexes satisfying the thresholding condition would indicate a single pothole. So, any continuous values are eliminated from the array as well, retaining only one of those values. The algorithm given below applies the final condition of Z-thresholding algorithm. This condition requires the previous & the next values of acceleration to be lesser than the values of acceleration at the index of the pothole.

```python
lat_final=[]
longi_final=[]
length=len(pothole_final)
index_final=[]
v=pothole_final[length-1]+1
pothole_final.append(v)
length=len(pothole_final)
for i in range (length):
    t=(pothole_final[i])-1
    if((df.iloc[t].z)<(df.iloc[t+1].z)):
        y=i
        #t1= (pothole_final[y+1])-pothole_final[y]
        while((pothole_final[y+1]-pothole_final[y])==1 and y+1<length-1):
            y=y+1
        #print(y)
        t=pothole_final[y]+1
        if((df.iloc[t].z<df.iloc[t-1].z)):
            lat_final.append(df.iloc[t-1].lat)
            longi_final.append(df.iloc[t-1].longi)
            index_final.append(t-1)

            #print('Pothole Detected')
pothole_final.pop()
index_pothole=list(set(index_final))
index_pothole.sort()
print(index_pothole)
```

```
[60, 65, 73, 83]
```

Fig 9: Indication the indices where potholes have been detected

The graph as shown in Figure 5, depicts the Z-axis acceleration with respect to time. It is evident from the above graph, that the Z-threshold algorithm accurately detects potholes, except in the case of one observation. The observation at time ~ 50, shows an observation which satisfies all the conditions of Z-thresholding algorithm. However, this observation should not be considered a Pothole. This is due to the fact that points, which were suspected Potholes had been grouped in the initial stage, and that the point satisfies the conditions of Pothole. This shows that not all the Predictions of the Z-threshold Algorithm are accurate. However, our analysis shows that the Z-threshold algorithm was able to accurately determine the coordinates of the Pothole in most of the cases.

7.2 Graphical Analysis

As soon as the indexes of pothole are determined using the Z-thresholding algorithm, as shown above, an SMS alert is sent to the authorities. A graph is plotted of the values of acceleration of Z-axis, the points where the Z-thresholding algorithm detects a pothole are also plotted & the points are connected in the form of a straight line.

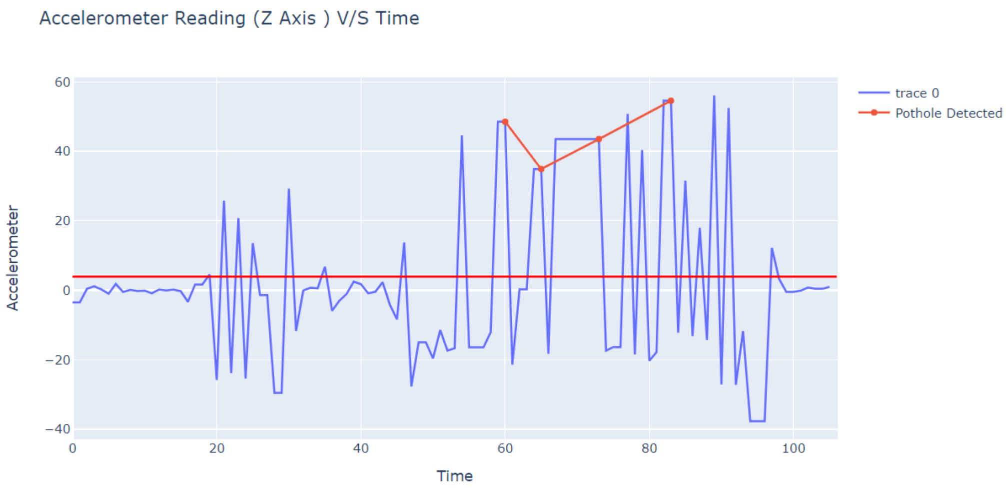

Fig 10: Indicating the points where potholes have been detected according to the Z-thresh algorithm

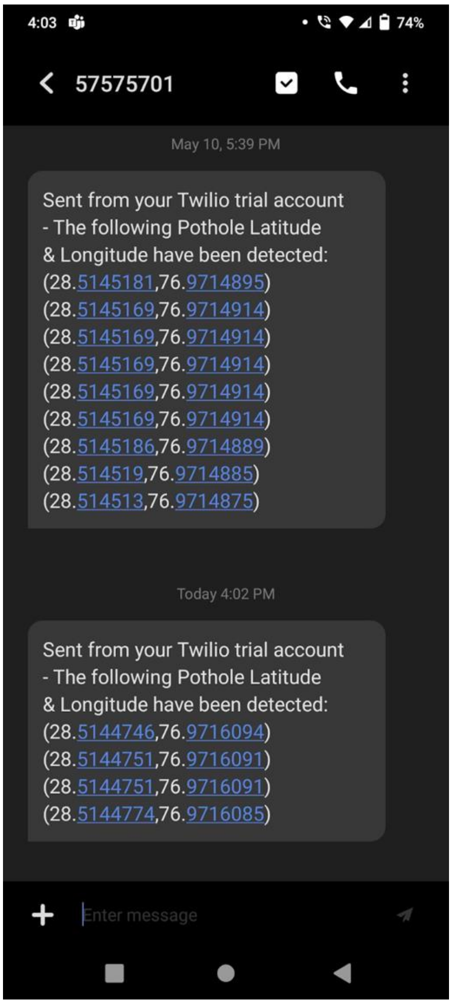

Fig 11: SMS received by the user with the location of the pothole detected

8. CONCLUSION

The first step in this project was building an application on Android Studio to detect potholes with the help of mobile's position which would detect potholes if there is a jerk.The values for x, y, z was computed along with the latitudes and longitudes. These data values were stored in firebase from Android. From firebase, we generated a json file which was converted to csv using python script. If the values of the z crossed certain threshold that is 3.92 (9.8*0.4) then there would be a pothole. To assure the pothole we checked whether the indices values were consecutive and if they were then the values of z were updated along with their latitude and longitude and these values were demarcated and shown that at these positions there lies a pothole. For this we use the Z-threshold algorithm.

Once the pothole is detected then SMS is triggered using Twilio alerting that there is a pothole. These alerts are useful for people commuting on a certain route and they should be careful when they go about a particular route having a pothole. Also, these alerts make road authorities aware of the fact that they need to repair these roads well in advance and make it comfortable for people to commute. Therefore, we believe that with our project we would be able to save people from road accidents and also the wear and tear of the tyres that frequently occur on the road.

We believe that our system would be useful in rainy areas as well areas of low visibility due to smog and fog thereby helping people commute at places they wish to.The concept of detection of Potholes can be extended to the following applications in the domain of smart city

1. **Informing Authorities:** Pothole detection triggers a new alert to a website, which can be used by the officials of the government, thus helping in the fast conveyance of information.
2. **Google Maps Integration:** A simple API call can help the millions of users across the globe to know about the Potholes around their area.
3. **Peer-to-Peer Smart Car Network:** In the future, the cars would communicate with the nearby smart cars using peer-to-peer networks, it helps ensure faster communication.

9. REFERENCES

[1] Agebure, M. A., Oyetunji, E. O., & Baagyere, E. Y. (2020). A three-tier road condition classification system using a spiking neural network model. *Journal of King Saud University-Computer and Information Sciences*.

[2] Harikrishnan, P. M., & Gopi, V. P. (2017). Vehicle vibration signal processing for road surface monitoring. *IEEE Sensors Journal*, *17*(16), 5192-5197.

[3] Carlos, M. R., Aragón, M. E., González, L. C., Escalante, H. J., & Martínez, F. (2018). Evaluation of detection approaches for road anomalies based on accelerometer readings—Addressing who's who. *IEEE Transactions on Intelligent Transportation Systems*, *19*(10), 3334-3343.

[4] Staniek, M. (2021). Road pavement condition diagnostics using smartphone-based data crowdsourcing in smart cities. *Journal of Traffic and Transportation Engineering (English Edition)*.

[5] Xu, Y., & Helal, A. (2015). Scalable cloud–sensor architecture for the Internet of Things. *IEEE Internet of Things Journal*, *3*(3), 285-298.