**Detection of Potholes Using Smartphone Sensors**

**SP19-CSCI-59000- 23627**

**Advanced Mobility and Cloud Computing**

**Project Report**

**Under the guidance of:**

**Dr. Arjan Durresi**

**By**

**Prem Chand Avanigadda**

**Department of computer and information science**

**Indiana University Purdue University Indianapolis**

**ABSTRACT:**

The aim of this project is to develop a mobile application to monitor potholes, they may cause severe damage to vehicles and lead to major accidents, In USA, they stand in 18th position in causing accidents. Keeping road safety in mind and acknowledging wide usage of smartphones, we proposed a real time monitoring system using smartphone sensors and machine learning algorithms, which detects and logs potholes data to AWS server.

**Keywords**: potholes, smartphone, sensors, machine learning algorithms, AWS server.

|  |  |  |
| --- | --- | --- |
|  | **CONTENTS** |  |
|  |  |  |
| 1. INTRODUCTION |  | 1 |
| 2. MOTIVATION |  | 1 |
| 3. RELATED WORK |  | 1 |
| 4. METHODOLOGY |  | 2 |
| 5. DESIGN PHASE |  | 3 |
| 6. PROGRAMMING PHASE |  | 5 |
| 7. RESULTS |  | 7 |
| 8. FUTURE WORK |  | 9 |
| 9. CONCLUSION |  | 9 |
| REFERENCES |  | 9 |

1. **INTRODUCTION**

A pothole [1] is a round or cylindrical depression on the surface of the roads, those depressions become large holes to become a pothole. Water and traffic are Main factors to cause potholes on the road surface. Water enters into cracks of pavement and weaken the soil beneath it, road traffic applies more pressure to week pavements results from depression on roads, these depressions may turn into worse by abrasion of traffic. So, it’s a widely natural phenomenon in frozen and frequent climate changing areas. Pothole may damage in a number of ways, firstly they cause accidents when car driver tries to avoid potholes by applying sudden breaks or taking the wrong turn. Secondly, they cause damage to the vehicle, including suspension failures, wheel damage, and tire punchers. In united states, only potholes estimate to damage about 3 billion dollars a year [1]. Potholes may cause server accidents, they stand 18th in the list of reasons for accidents. Alone in India [1], 3000 people are killed in accidents caused by potholes. they became more frequent and causing a lot to damage to economy and people.

As pothole formation is natural phenomena, finding them, reporting and repairing are the main issues in front of the governments. Everything should be done manually, which costs millions to state governments. In this project, we propose a system to do all the things mentioned above. In 2015, Canada spent 4.8 million dollars [1] to report 4,50,000 potholes manually. So, even monitoring is a big task in front of societies to avoid accidents and damage caused by potholes. We tried to answer some of the issues like finding and reporting a pothole in our proposed system.

Duration of the pothole increases the damage, timely reporting them helps the city management to repair them. So, quick reporting may reduce severe damages to the car or may avoid accidents. This android application automatically monitors the potholes and report them to the authorities and warn the drivers on the road.

1. **MOTIVATION**

Almost everyone had a smartphone in their hand, its wide spread usage and high computing power are useful to solve some issues in society. Its costly and infeasible solution to develop a separate embedded system to detect potholes, where mobiles come with inbuilt sensors, so its best way to utilize them, so we decided to develop a mobile application to monitor potholes.

1. **RELATED WORK**

There are few papers described the usage of acetometers in their sensing systems. We will see some of them get ideas to design our own system.

In Boston, researchers from MIT developed a Pothole Patrol system [2], their system built on machine learning algorithms, they collected X, Z values and velocity from external acetometer, location co-ordinates from an external GPS device. Filtered the data through five filters namely speed, high pass, z-peak values, ratios of xz values and z ratio vs speed. They employed machine learning in last three filters. Disadvantages of this approach are using external hardware and it is not a real-time system.

The University of Colombo developed BusNet [3] system, they used MICAZ microcontroller and external sensors boards to get acceleration and GPS coordinates. It is not real time, They used this device only for the collection of data and reporting it in local bus stations due to less local storage.

Microsoft research group in India developed a pothole detection system namely Nericell [4] and TraffocSence [5] using mobiles with windows OS. They used external hardware like microphone and GPS. They used simple Threshold based algorithms Z-peak and Z-sus along with virtual reorientation algorithm to nullify the effect of mobile orientations during driving.

National Taiwan University [6] developed a client-server model to detect potholes, on client side they sued HTC mobile having built-in acetometer and external hardware to get GPS coordinates. They filtered and preprocessed data before sending it to the server. On the server side, they employed two machine learning models namely SVM and smooth road model to predict the pothole by taking client data as input. Advantage of this model is being a real-time system, using the built-in accelerometer. Disadvantages are they used an external GPS device to get location coordinates.

We took some of the useful features and methods from the above sensing systems, we will see our proposed methodology in the next section.

1. **METHODOLOGY**

We will take some useful features from previous models such as client and server architecture, using high pass filters to remove unnecessary logs while monitoring, and usage of machine learning models. The core idea of our sensing system is to avoid external sensors. Now a day’s smartphones are equipped with an accelerometer, gyroscope, linear accelerometer, GPS, magnetometer and barometer, etc. Usually, external accelerometers and built-in accelerometers give readings affected by gravity. We need to preprocess the readings to remove the effect of gravity, which affects the efficacy of the system. So, we will use the linear accelerometer to get accurate x, y, z values without the effect of gravity “g”.

We will use a high pass filter to remove small accelerations caused by physical movements of the human body. We will use z and x thresholds filters to remove unnecessary logs.

Then obtained logs are sent to the server to predict whether it is a pothole or not by machine learning models. The obtained pothole log is stored in a database for future usage.

A close up of a logo

Description automatically generated

Fig 1: sensor data flow

We can observe from Figure 1 that Our approach is real-time monitoring, sensor data is collected from the smartphones and passed through the various filters at the client side, sensor data is given as input to the machine learning model at server side. Based on the decision of our trained model, it is stored in the database.

1. **DESIGN PHASE**

In a client-server architecture, it takes long development time to develop everything from scratch on server components like REST, database, and a common platform to run all these services. So, we take advantage of using AWS cloud services to reduce development time and make use of efficient virtual cloud platforms.

As shown in Figure 2, we will use API GATEWAY, Lambda services, IAM role, and DynamoDB. Let’s discuss each in detail.

* API GATEWAY: Amazon provides full packaged service for developers, we can create, deploy and monitor REST applications in a few minutes. They are connected to other services like Lambda and IAM role.
* AWS Lambda: it comes with various programming environments; we can access any service in AWS through lambda function. The example we can connect to S3 storage, DynamoDB, elastic services, and multiple AWS services with few clicks.
* IAM role: It manages the security aspect of all AWS services, we can create user roles, policies using IAM role and assign them to different AWS services to restrict their access and management.
* DynamoDB: it is key-value document database, it can handle millions of requests per second. It has a dynamic schema, so table schema can be changed in runtime.

**5.1 Procedure:**

Inbuilt Accelerometer and GPS collects the sensor data from the smartphone and passes through the Threshold filters, if the X and Z values are more than the threshold values, then data is sent to REST API present in the API GATEWAY, REST API sends the data to lambda function, the machine learning model in lambda function evaluates the input and decides whether it is pothole or not. If it is a pothole, then log data is stored in the DynamoDB.

**A screenshot of a cell phone

Description automatically generated**

Fig: 2 pothole detection architecture [7]

**5.2 Design problems on the server side:**

In our system, we face problems onthe server side with handling multiple requests from the client, so we decided to develop a Representational state transfer application also known as REST API. As of now, we are not getting any data from the server in mobile, just sending the logs to the server, so we used API GATEWAY to create a POST method to receive data from the mobile application.

We faced another design problem while saving logs to DynamoDB. It started throwing floating point errors for X, Y, Z values, it only accepts decimals and strings.so, we converted the float values to decimal values in lambda function. Another design problem is to set the key for sensor data in DynamoDB. For example, if we pick timestamp as key in the database, then two devices send data at the same time, we will get a primary key error, in another case, when we pick GPS co-ordinates as primary key, then a device sends the same location already present in the database leads to primary key error. So, we decided to create a unique value by missing device UUID and timestamp. Which will be unique in all cases.

* 1. **Design problems on the client side:**

While designing Z and X threshold filters, we faced a problem while defining a common ideal value for all road surfaces, usually, it depends on the nature of the road surface. When the road is smooth, we can detect potholes at normal thresholds, but when the road surface is not good, we will get high values, which can pass filters and send to server, which reduces the system performance. So, we added dynamic range sliders in the mobile application, to take the custom X and Z thresholds form the user.

1. **PROGRAMMING PHASE**

**6.1. IONIC platform and framework**

IONIC [8] is used to develop cross-platform mobile applications, we can deploy our application to IOS, Android or Windows platforms. We use HTML, CSS, Angular.js, and Typescript to write code for mobile application. The IONIC framework comes with pre-designed components, we used ion-button, ion-ranger in our mobile User interface.

**6.2. Plugins**

Ionic has a lot of native Cordova plugins, we can use them to save our development time, let's see some of the important plugins we used in development.

* **cordova-plugin-geolocation 4.0.1 "Geolocation*"***: it is used to get device location coordinates such as latitude and longitude.
* **cordova-plugin-sensors 0.7.0 "Sensors":**  it is used to get device linear acceleration, which includes X, Y and Z values.
* **cordova-plugin-device 2.0.2 "Device":** It is used to get device information like UUID, platform type, etc.
* **cordova-plugin-advanced-HTTP 2.0.9 "Advanced HTTP plugin":** It is used to make HTTP requests to the server, we used it to post the data to AWS RESTFUL Gateway.
* **cordova-plugin-splashscreen 5.0.2 "Splashscreen":** It is used to add a splash screen to the mobile application.
* **cordova-plugin-whitelist 1.3.3 "Whitelist":** It is used for device navigation and content security.

These are some of the important plugins, we used in our application development.

**6.3. Machine learning classifier**

We got the pothole data from CRAWDAD [9] group at Dartmouth University, in preprocessing, we merged pothole data from multiple files into one and remove the missing data rows. Removed the timestamp, speed breaker and location co-ordinates form the pothole data. Chosen python as a programming language to develop a machine learning model. Split the data in 80:20 ratio to get train and test data respectively. Used Sklearn python library to build a model and saved as “pothole.pkl” format using pickle libraries.

**6.4. Deploying the ML model to AWS lambda**

We uploaded the “pothole.pkl” file to S3 buckets and access the machine learning model from lambda function. In lambda environment, we don’t have the sklearn, numpy python libraries. So, we added one extra layer called “AWSLambda-Python37-SciPy1x” contains numpy library as shown in Fig 5. For sklearn and numpy library, we need to zip the sklearn and numpy executables along with “lambda\_function.py” and upload the .zip folder to the AWS Lambda.

1. **RESULTS**

**7.1 Android application Screens:**

Tested application on android device.

**A screenshot of a cell phone

Description automatically generated**

Fig 3: Pothole mobile application screens

**7.2 AWS API GATEWAY**

Created a “POST” method in “mypothole-api” application

A screenshot of a cell phone

Description automatically generated

Fig 4: POST method in REST API

**7.3 AWS Lambda architecture:**

Lambda function “valiadtePothole” having extra layer to use python libraries.

A screenshot of a cell phone

Description automatically generated

Fig 5: lambda architecture

**7.4 machine learning model:**

We got the accuracy of model as 96 percentage. Precision of 0.96, recall 0.96 and f-score is 0.96.

**A screenshot of a cell phone

Description automatically generated**

Fig 6: results of the ML model

1. **FUTURE WORK**

we can add gyroscope to the application to detect mobile orientations so that we can filter more unnecessary logs. By combining gyroscope and linear acceleration results, we will get accurate results. We can add other features on the server side like reporting to the respective authorities, mapping on google maps, tweeting about the pothole.

**9. CONCLUSION**

This project is to use limited hardware to detect potholes while driving, we abled to design and create a client server architecture and android application for real-time monitoring. Usage of this application may reduce the damage of potholes on the economy and creates a monitoring system for the government to take necessary steps.

**REFERENCES**

1. En.wikipedia.org. (2019). Pothole. [online] Available at: https://en.wikipedia.org/wiki/Pothole [Accessed 29 Apr. 2019].
2. Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S. and Balakrishnan, H. (2008). The pothole patrol. Proceeding of the 6th international conference on Mobile systems, applications, and services - MobiSys '08.
3. De Zoysa, K., Keppitiyagama, C., Seneviratne, G. and Shihan, W. (2007). A public transport system based sensor network for road surface condition monitoring. Proceedings of the 2007 workshop on Networked systems for developing regions - NSDR '07.
4. Mohan, P., Padmanabhan, V. and Ramjee, R. (2008). Nericell. Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys '08.
5. Mohan, P., Padmanabhan, V., Ramjee, R. and Padmanabhan, V. (2019). TrafficSense: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones. [online] Microsoft Research. Available at: https://www.microsoft.com/en-us/research/publication/trafficsense-rich-monitoring-of-road-and-traffic-conditions-using-mobile-smartphones/ [Accessed 29 Apr. 2019].
6. Tai, Yu-chin & Chan, Cheng-wei & Hsu, Jane. (2010). Automatic road anomaly detection using smart mobile device.
7. freeCodeCamp.org. (2019). How to quickly create a serverless RESTful API with Node.js and AWS. [online] Available at: https://medium.freecodecamp.org/quickly-create-a-serverless-restful-api-with-nodejs-and-aws-lambda-api-gateway-and-a6be891cc16a [Accessed 29 Apr. 2019].
8. Ionicframework.com. (2019). Ionic Documentation. [online] Available at: https://ionicframework.com/docs [Accessed 29 Apr. 2019].
9. Mohit Jain, Ajeet Pal Singh, Soshant Bali, Sanjit Kaul, CRAWDAD dataset jiit/accelerometer (v. 2012‑11‑03), downloaded from https://crawdad.org/jiit/accelerometer/20121103, https://doi.org/10.15783/C76K58, Nov 2012.a