Intelligent Pothole Detection and Road Condition Assessment using Smartphone Sensors

Umang Bhatt
Electrical & Computer Eng.
Carnegie Mellon University
Pittsburgh, PA 15213
umang@cmu.edu

Shouvik Mani
Statistics & Machine Learning
Carnegie Mellon University
Pittsburgh, PA 15213
shouvikm@cmu.edu

Edgar Xi
Statistics & Machine Learning
Carnegie Mellon University
Pittsburgh, PA 15213
esx@andrew.cmu.edu

Zico Kolter
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
zkolter@cs.cmu.edu

Abstract

We propose an intelligent system to assess road conditions and detect potholes using smartphone gyroscope and accelerometer sensors. After collecting over 21,000 sensor observations, transforming the data into useful signals, and training a series of classifiers, we demonstrate the successful classification of road conditions. This approach is then modified to perform the detection of potholes. We present a publicly available dataset for road condition and pothole classification, as well as a proof of concept of a real-time classification smartphone application for this purpose.

Introduction

Potholes and poor road conditions are a nuisance to the public, leading to discomfort, vehicle damage, and accidents. Unfortunately, they are also fairly common and go ignored too often. Civic authorities are not always aware of potholes, and road repairs happen only intermittently. We believe that we can tackle these issues by developing a system that uses smartphone sensors to classify road conditions and potholes in real-time and alerts authorities about roads that are in need of repair.

Several papers have successfully used smartphone accelerometer sensors to detect potholes. In our paper, we present a novel approach of using a combination of gyroscope and accelerometer sensors for pothole detection, as well as assessing overall road conditions. We built a classifier that is capable of real-time pothole classification, and we are releasing the application and the data to the general public, so that intelligent pothole detection can be crowd sourced.

We use an iPhone's accelerometer and gyroscope sensors to provide insight into the condition of the road being traveled on. An accelerometer tells us the linear acceleration in the X, Y, and Z directions, while the gyroscope can detect rotational motion in all three directions. Enumerating the linear and angular movement of the phone via these two sensors, we want to address two central tasks.

1. Differentiate between good and bad road conditions.
2. Detect potholes in real-time.

Related Work

Several other papers have demonstrated the use of smartphone accelerometer data to classify potholes as well as assess road conditions, but our classification task differs from others in its inclusion of gyroscope data. These papers are outlined briefly below.

Mednis, et al demonstrate in their paper "Real time pothole detection using Android smartphones with accelerometers" that android smartphones can be used to detect pothole events. Using a classification scheme that labels accelerometer activity that crosses a certain z-axis threshold, with an optimal value of 0.4g, as well as a difference in z-axis with optimal value 0.2g, their algorithm detects potholes with true positive rates as high as 90%.[1]

P Mohan. et al present Nericell, using a fleet of smartphones to assess road conditions and an aggregation server, as well as a set of algorithms to reorient a disoriented smartphone accelerometer along a canonical set of axes.[2]

Eriksson, Jakob, et al use a crowdsourced fleet of taxis called Pothole Patrol, gathering accelerometer and GPS data to identify potholes and road anomalies with a misidentification rate of 0.2%.[4]

Methodology

Before classifying potholes and road conditions, we had to collect a considerable amount of training data. We built a system for collecting and labeling this data via two separate iPhone apps. Since the performance of the classifiers would be limited by the quality of training data, much emphasis was placed on establishing appropriate controls and ensuring reproducibility. In this way, we could isolate the effect of the accelerometer and gyroscope sensors on the corresponding label, instead of having other confounding variables get in the way. Finally, we applied various transformations and aggregations on the raw sensor data to get a better signal for classification.

Specifications

All of the data was collected using a 2007 Toyota Prius with approximately 100,000 miles. Both of the smartphones used for data collection were iPhone 6Ss. One iPhone was used for collecting sensor data while the other for recording potholes. An iPhone suction-cup mount was used to place the iPhone collecting sensor data on the center of the windshield.

Variable Definition and Controls

For both of the problems, we needed to establish test groups and control for confounding variables. In the good road/bad road problem, we reduced the varying degrees of road conditions to two extremes. We did multiple drives on poor quality roads and on good, straight roads. There was no pothole annotation done on these routes. We simply wanted to see if gyroscope and accelerometer sensors could effectively differentiate between a good and bad road.

For the pothole detection problem, a major confounding variable was the route used for data collection. Different routes could have varying numbers and quality of potholes. In order to control for this and ensure reproducible results, we decided to collect data on a single route. This route had a mix of pothole-free and pothole-filled stretches and ensured that we produced a balanced dataset. We traversed the route, shown in Figure 1, in only one direction.

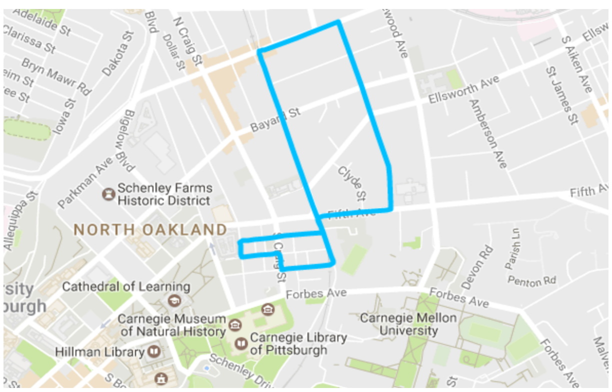

Figure 1: Route for collecting training data on potholes

Data Collection

To facilitate the collection of training data, we built two iOS applications. One app collected sensor data (Figure 2). Specifically, five times per second, it recorded a UNIX timestamp, accelerometer data (x, y, z), gyroscope data (x, y, and z), location data (latitude and longitude), and speed. This app was run on an iPhone mounted near the center of the windshield of the car. It was used for both the good road/bad road and pothole detection problems, since both needed features on the car's movement.

The second app (Figure 3) was used to annotate when a pothole was driven over - ideally, we wanted to get the exact time when a pothole was hit, but we will later discuss how we accounted for human error. This app was run on an iPhone given to a person on the passenger-side, whose job was to label the potholes. The passenger would simply click a button when he or she felt a pothole, and the UNIX timestamp would be recorded. This app was used for the pothole detection problem and was run alongside the other iPhone collecting sensor data.

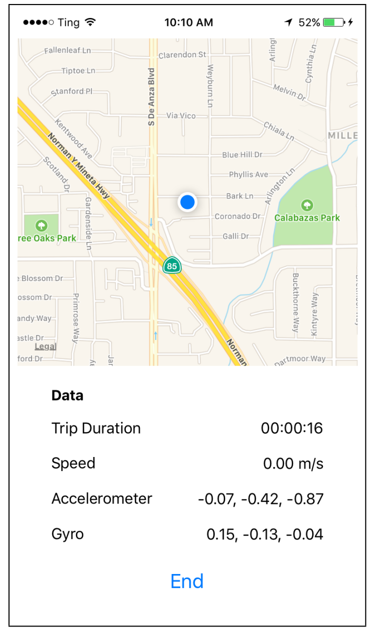

Figure 2:

App 1 collected sensor data (timestamp, accelerometer, gyroscope, location, and speed)

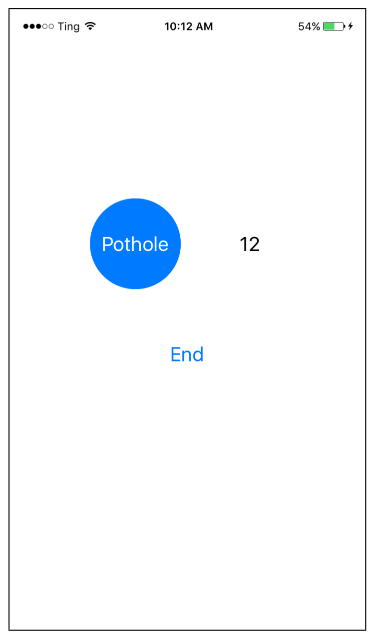

Figure 3:

App 2 was used for labeling potholes and their timestamps

To minimize undesired variance in our data collection, we set some controls. We used only one driver and one pothole recorder throughout the entire data collection process.

Feature Engineering

Once the individual training datasets (sensor data and pothole labels) were collected and combined, we had over 21,300 observations (71 minutes) of raw accelerometer and gyroscope readings as well as 96 labeled potholes. But since the sensor data was collected at a high frequency of 5 times per second, it was likely that the sensors captured some movements unrelated to vibrations caused by road conditions. So, the individual sensor data points were noisy and did not capture our variables of interest.

To resolve this issue, we grouped data points into intervals and calculated aggregate features for each interval from the individual features. We created a set of 26 aggregate features for each interval which included:

- Mean accelerometer x, y, z
- Mean gyroscope x, y, z
- Mean speed
- Standard deviation accelerometer x, y, z
- Standard deviation gyroscope x, y, z
- Standard deviation speed
- Max accelerometer x, y, z
- Max gyroscope x, y, z
- Min accelerometer x, y, z
- Min gyroscope x, y, z

Note: Aggregates for x, y, z dimensions for accelerometer and gyroscope sensors are three separate features.

For road condition classification, we decided to use an interval of 25 data points (5 seconds). We believed that 5 seconds was ample time to assess a small stretch of a road and classify it as good or bad. After creating the 5-second intervals and aggregate metrics, we attached the corresponding labels of good road (0) and bad road(1).

For pothole classification, we used an interval of 10 data points (2 seconds). Since potholes are sudden events, we hypothesized that a shorter interval would be able to capture them more accurately. For each interval, we attached the corresponding label of non-pothole (0) or pothole (1), depending on whether a pothole occurred during that interval.

Stitching together the sensor data and the labeled pothole data was a non-trivial problem, since labeling the potholes was itself an error-prone task. A person labeling potholes could be too late in clicking the pothole button or may click the button accidentally. By grouping the points into intervals, we addressed the former since a person could be slightly late in clicking the button but that interval would still be labeled as a ”pothole” interval.

Data Exploration

We started by visualizing the data we gathered to see if we ourselves could notice any patterns. Then, and only then, would we be able to build useful classifiers. The goal of the following figures is to understand the data and come to meaningful conclusions that we can then transfer to our classifiers.

Time Series

Time series plots helped us understand whether there was a benefit in using the intervals and aggregate metrics instead of individual data points. Figure 4 shows the comparison between a good road and bad road using individual accelerometer readings (centered). Although there is clearly a difference between the plots, with the bad road plot having a higher variance, both of the datasets are noisy.

In contrast, Figure 5 shows the same data, but grouped into intervals with their standard deviation accelerometer aggregates. Now, the difference between the good road and bad road data looks more pronounced. Doing this aggregation extracts the signal from the noise and produces a more stable set of features to use in our classifier.

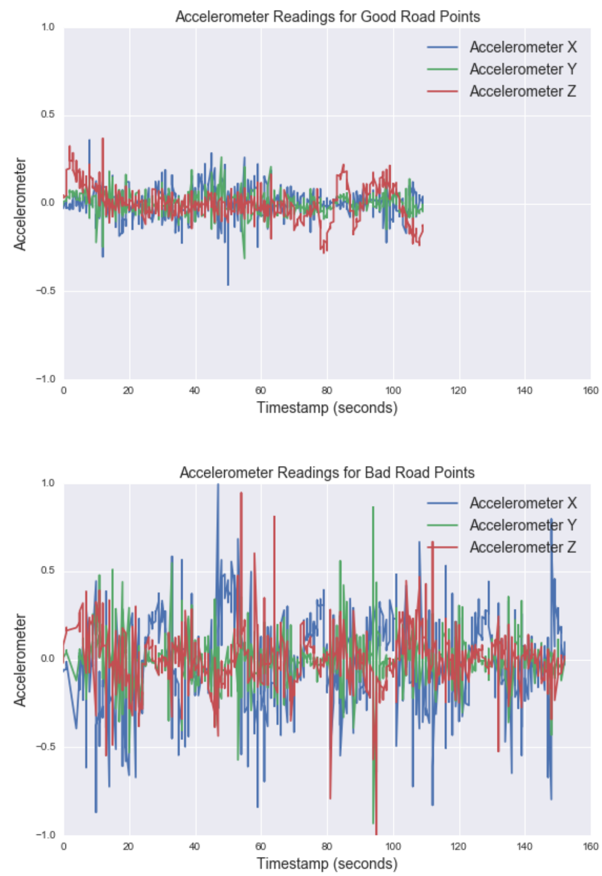

Figure 4: Accelerometer readings (centered) for good vs. bad road

Road Conditions Data Exploration

Since we were not using time as a feature, we created 3D point clouds to visualize the data independent of time. By running principal component analysis (PCA), we reduced the 26-dimensional feature space of the intervals into three dimensions. Upon plotting the intervals and coloring them by their road condition label, we found a clear linear separation between good road and bad road intervals, as seen in Figure 6.

Pothole Data Exploration

Similarly, we ran PCA on the pothole data and plotted and colored the intervals in their reduced three dimensions. Once again, we observed a linear separation between the pothole and non-pothole intervals, as seen in Figure 7.

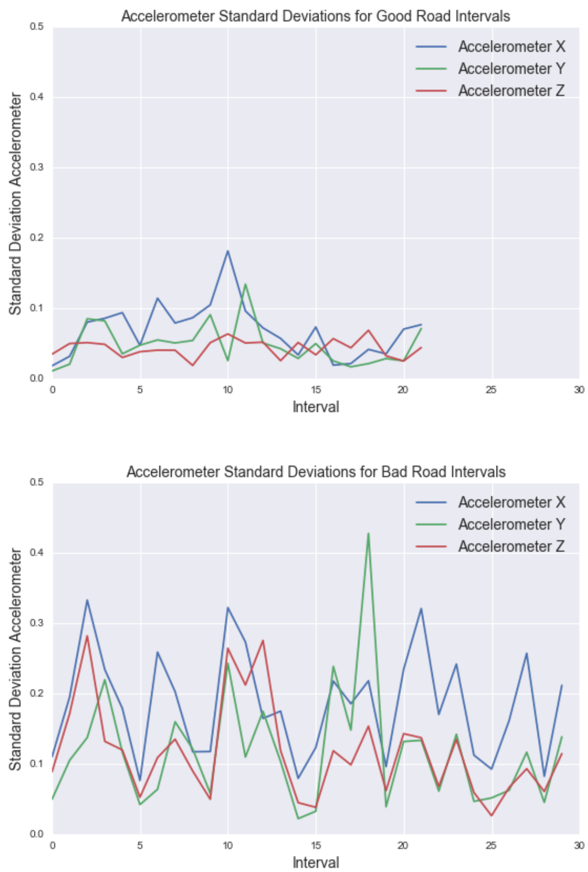

Figure 5: Standard Deviation of accelerometer readings for good vs. bad roads

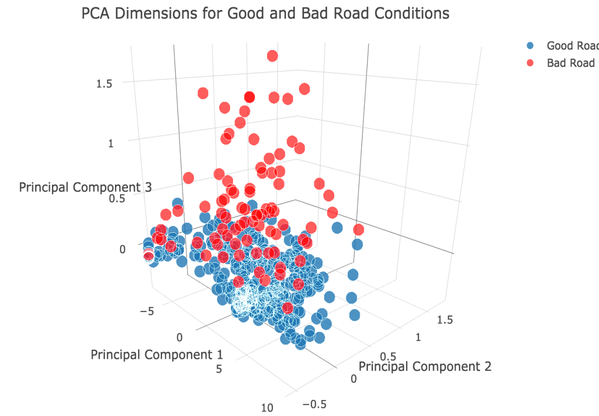

Figure 6: PCA of features colored by road conditions

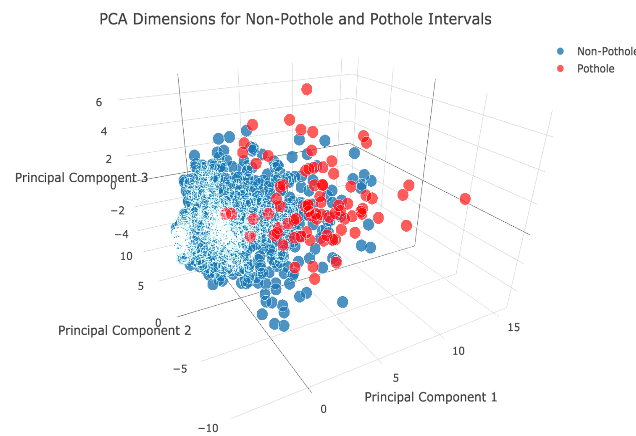

Figure 7: PCA of features colored by presence of pothole

Results and Analysis

After combining data from all of our data collection trips and generating intervals and aggregate metrics, we trained several classifiers for both of the classification tasks.

Road Condition Classification

The road condition dataset was used to trained and evaluate several classifiers, including support vector machine (SVM), logistic regression, random forest, and gradient boosting. The best results of each classifier can be found in the Appendix: Table 1. We tuned some of the parameters and hyperparameters for each classifier to get the best test set accuracy.

Overall, an SVM with a radial basis function (RBF) kernel and gradient boosting both achieved the highest test accuracy of 93.4%. A baseline model that predicted "good road" for all instances would have achieved 82% accuracy. SVM and gradient boosting did considerably better than this base rate and are useful classifiers in this problem.

Figure 8 illustrates the selection of the regularization parameter C for the SVM classifier. Note that the training and test error are fairly close to each other at the chosen parameter value C=250, indicating that the model is performing well. However, more data and more useful features could be helpful in further lowering this gap between the training and test error.

Pothole Classification

Since potholes are rare events, there was a large class imbalance in our dataset. Even a naive model that always predicted "non-pothole" for a new interval would achieve 89.8% accuracy on the classification task. So, in this problem, it was more important to optimize the precision-recall tradeoff than to focus on accuracy alone.

Like in road conditions classification, the best performing classifiers were SVM and gradient boosting, with accuracies of 92.9% and 92.02% respectively. These accuracies were somewhat better than the base rate of 89.8% from the baseline classifier, so at least the classifiers were useful.

The SVM model performed the best in terms of accuracy, but we wanted to improve its precision-recall tradeoff. The precision-recall curve in Figure 9 illustrates all the combinations of precision and recall values for different thresholds

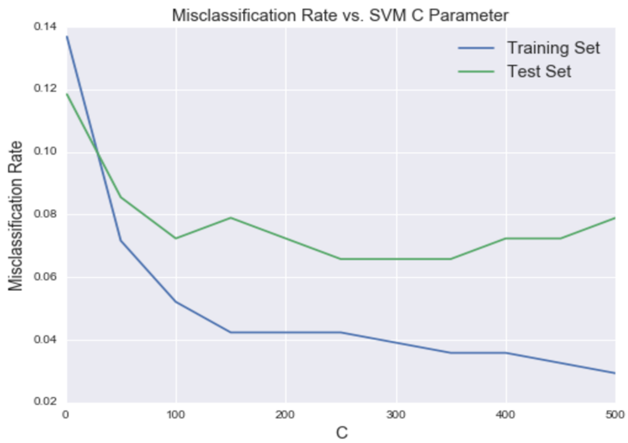

Figure 8: Optimizing SVM regularization parmeter

on the SVM decision function. The red point in the figure represents the threshold we chose which gives us a precision of 0.78 and a recall of 0.42.

This choice is a good tradeoff between correctly flagging potholes (high precision) and detecting all true potholes (high recall). In this context, a precision of 0.78 means that when our model classifies an interval as having potholes, 78% of those intervals actually have potholes. A recall of 0.42 means that our model correctly classifies 42% of the true pothole intervals. Notably, the accuracy of the SVM model stayed at 92.9% even though we changed the classification threshold to improve the precision-recall tradeoff.

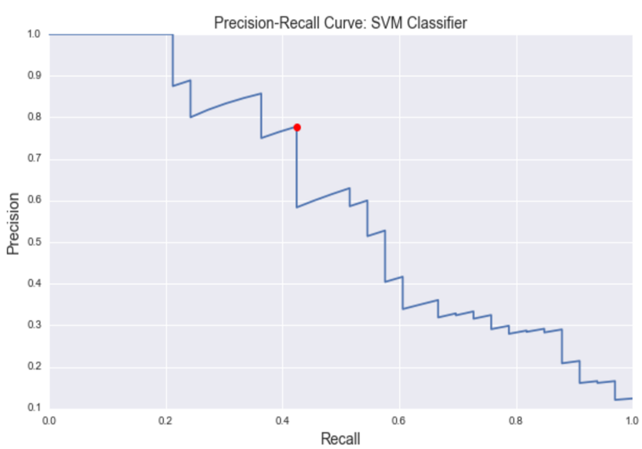

Figure 9: Precision-recall curve for the SVM classifier. The red point is the precision-recall tradeoff we chose.

Discussion

In this report, we have presented a publicly available dataset and examples of using this data for road condition and pothole classification. This data and classification groundwork lays the foundation for real-time classification applications that have a high social impact. Additionally, it offers lessons for doing such work in the future as well as extension points to improve the work we have done.

Real Time Classification Application

Having successfully built viable classifiers for both good road/bad road classification and pothole detection, we are currently developing a third iPhone app that does real-time classification for these tasks. While the previous apps were developed to collect training data, this app could be deployed to users on a large scale to assess road conditions and detect potholes ”in the wild.” The classification logic (i.e. a fitted SVM model) will be hosted on a cloud-based web server, and the app will be an interface for interacting with that classifier. Using crowdsourced data from devices running the application, we would be able to produce data-rich maps labeled with potholes and color-coded with road conditions from the entire city.

Social Good Application

Crowdsourcing the classification and detection of road conditions and potholes could significantly improve the maintenance of road infrastructure in our cities. One could imagine the real-time classification app mentioned above being deployed to thousands of devices, constantly collecting road condition and pothole data from across a city. This data could be shared openly and combined with insightful road condition maps to help public works departments direct road maintenance resources to where they are most needed.

According to Christoph Mertz, Chief Scientist of Roadbotics, smartphone sensors could also be put on public vehicles such as garbage trucks and post office vans, which cover the majority of a city’s road network. The ability to create a less invasive method of detecting potholes and classifying roadside conditions would make it easier to disseminate the smartphone app, allowing for the creation of a more detailed map of a city’s roadside conditions and potholes. We hope that our work provides a basis for further work in crouwdsourced public service.

Failures

While performing this data exploration and analysis, we ran into many bumps (no pun intended) and had to pivot our approach and methodologies. Below is a collection of the failures we had to overcome in order to produce safe and sound results for this project.

- From the inception of this project, we intended on building a classifier that works on all roads. Unfortunately, in order to build a reliable classifier, we would need sufficient data from roads of all types, which would take far longer than three months. Thus, we found it essential to select a specific route to classify on. Sticking to one route ensures that we collect enough data for a reliable classifier of that particular route.
- Before tackling the precise classification of individual potholes, we found it helpful to understand road conditions. We wanted to answer the question: can we differentiate between a *good* road and a *bad* road? After proving

the feasibility and the accuracy of road condition analysis, we felt comfortable and confident in moving to pothole classification.

- A major pivot point for us came when we divided our app into two separate apps. Instead of collecting sensor data and tagging potholes on the same phone, we had one app mounted onto the dashboard collecting data undisturbed, and the second app was given to the passenger who solely annotated when the car ran over a pothole.
- On multiple occasions, we lost our collected sensor data due to a lack of robustness in our original application, which needed to be loaded onto the phones once every two weeks. Had we done this project again, we would have invested time upfront into the applications to ensure they are reliable and robust during data collection.
- Since we are collecting sensor data five times per second, we ended up collecting over a thousand data points per time we traversed our route. When beginning our analysis on all our data from a given set of trips, we began to get bogus results; perplexed by what was happening under the hood, we quickly realized that trying to classify if a given fifth of a second occurs during a pothole is not insightful. Creating ten second intervals for detecting road condition and two second intervals for pothole detection proved to be more eye-opening. Over those intervals we were able to extract a multitude of features discussed above.

Future Work

There are many extension points from this initial data exploration and classification project. Below are a few examples of possible future work.

- Expanding the route to collect training data on additional roads could only help by decreasing the variance of the models.
- Building a device to solely capture accelerometer and gyroscopic data (though this does violate the aforementioned *invasiveness*) would allow for less variance due to confounding variables.
- Working to control other confounding variables, like sudden changes in acceleration or mere braking, would make the features of the classifiers more robust.
- Calculating road condition scores (from 1-10, per se) would help extend this project beyond binary classification. These scores can then be mapped onto a given city/route to denote the conditions of roads comparative to other roads featured on the given map.

Acknowledgments

In addition to continual guidance and encouragement from Professor Zico Kolter of Carnegie Mellon University's School of Computer Science, we are also especially grateful to Professor Christoph Mertz for his insight to use gyroscope and accelerometer data to classify potholes. We would also like to thank Professor Roy Maxion, Professor Max G'sell, and Professor David O'Hallaron for their patient and thoughtful advice.

References

1. A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs and L. Selavo, "Real time pothole detection using Android smartphones with accelerometers," 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS), Barcelona, 2011, pp. 1-6.
2. P. Mohan, V. Padmanabhan, R. Ramjee, Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones, Microsoft Research India, Bangalore, India, 2008
3. Mohan, Prashanth, Venkata N. Padmanabhan, and Ramachandran Ramjee. "Nericell: rich monitoring of road and traffic conditions using mobile smartphones." Proceedings of the 6th ACM conference on Embedded network sensor systems. ACM, 2008.
4. Eriksson, Jakob, et al. "The pothole patrol: using a mobile sensor network for road surface monitoring." Proceedings of the 6th international conference on Mobile systems, applications, and services. ACM, 2008.

Appendix

Table 1: Road Conditions Classification Metrics

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Baseline	0.82	0	0	0
SVM	0.93	0.79	0.85	0.82
Logistic Regression	0.92	0.68	0.39	0.50
Random Forest	0.92	0.83	0.72	0.78
Gradient Boosting	0.93	0.90	0.70	0.79

Table 2: Pothole Classification Metrics

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Baseline	0.89	0	0	0
SVM	0.92	0.78	.42	0.55
Logistic Regression	0.92	0.68	0.39	0.50
Random Forest	0.92	0.75	0.36	0.49
Gradient Boosting	0.92	0.65	0.45	0.54