INFLUENCE OF VEHICLE MAKE ON ACCURACY OF REAL-TIME ROAD ANOMALY IDENTIFICATION

An Undergraduate Research Scholars Thesis

by

MICHAEL TAYLOR SAULNIER

Submitted to the Undergraduate Research Scholars program at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Research Advisor: Dr. Jim Ji

May 2018

Major: Electrical Engineering

TABLE OF CONTENTS

ABSTRACT

Influence of Vehicle Make on Accuracy of Real-Time Road Anomaly Identification

Michael Taylor Saulnier
Department of Electrical and Computer Engineering
Texas A&M University

Research Advisor: Dr. Jim Ji
Department of Electrical and Computer Engineering
Texas A&M University

As road infrastructure in the United States is aging, road anomalies such as cracks, potholes, and other abnormalities are becoming much more prevalent. Currently there is no real-time understanding of the conditions of roads, thus we developed a machine-learning algorithm developed and trained to identify road conditions in real time based on data collected by smartphones. Since there are a multitude of different vehicles on the roads and locations where phones can be placed in the vehicle, creating a classification algorithm that can work regardless of the vehicle type and phone placement is incredibly important. Doing a comparative study on the different vibrations received at different locations in different vehicles will provide a baseline for future development of a universal algorithm that uses crowd sourced data from cell phones to allow for real-time awareness of changing road conditions. This in turn provides a way to identify and fix dangerous road anomalies quickly.

ACKNOWLEDGEMENTS

I would like to thank my mentor, Dr. Jim Ji, for his guidance and support throughout the course of this research. His patience and willingness to provide insight was key in the completion of this project. From a study abroad, to the classroom, and now to the lab, Dr. Ji has been instrumental in my development as a student, and I am forever grateful for my time with him at Texas A&M.

I would also like to thank Akanksh Basavaraj for creating the IOS data collection program and his support and expertise throughout the entire process. Dillon Knox for his support through the machine learning algorithm development and work during the data collection stage of this project.

Many thanks to Dr. Jing Du of the Texas A&M Construction Science Department for providing the Raspberry PI and DJI Video camera used during the data collection stages as well as his expertise in road anomalies and the road construction process.

Finally, thanks to my mother and father for their encouragement and support throughout the entirety of college.

CHAPTER I

INTRODUCTION

Background

Road anomalies can take the form of cracks, potholes, or numerous other road abnormalities. As road infrastructure in the United States has aged, anomalies are becoming much more prevalent. Anomalies can be incredibly dangerous and can cause accidents. They can also cause lower fuel efficiency and wear and tear on vehicles. Currently road monitoring requires expensive and complicated vehicular instruments, such as laser profilometers to calculate international roughness index or using ground penetrating radar to determine the condition of the road (Cao, Labuz, & Guzina 2011). These methods are incredibly costly, time, and man intensive because of how specialized the instrumentation is. Many times smaller municipalities and local governments simply cannot afford these methods of monitoring road conditions. Since many people have smartphones in their cars, it would be ideal to use the smartphones to in an attempt to classify and provide a location for these dangerous anomalies. While developing an application to collect data from smart phones has been done in the past, this project will focus on the comparison of pothole detection between two different vehicles. This project provides groundwork for a larger system that would allow for cloud sourced road information. This information could save massive amounts of money not only on expensive equipment and man-hours, but also allow for planners to use money more efficiently with real-time knowledge regarding the pace of road deterioration. This technology could potentially be further monetized by integrating the algorithm into an application like Google Maps, that would find not only the quickest route, but also the smoothest ride.

Related Work

There has been a substantial amount of work done to try and characterize roads and pavement using smart phones and other inexpensive technologies. Initially there has been work that showed that it would be possible if there was a pot hole or speed bump by thresholding accelerometer data along the z-axis (Kalra et al. 2014). Unfortunately this is substantially limited by the setup, which controlled variables precisely and smartphone orientation. Another avenue of research has been not using smartphones to collect data. Another project used Arduinos to capture the data, and then mapped that data to Google Earth, which provided a high sampling rate with accurate data and also an accurate location for the road anomalies (Chen et al. 2015). Using Arduinos would be difficult to eventually crowd source because of the specificity of the setup, which is why we will be using smartphones.

Another project did a substantial amount of work attempting to characterize road roughness index using SmartRoadSense, a crowd sourced application that can be downloaded onto smart phones and transmit GPS coordinates, vehicle speed and roughness index to a common database (Alessandroni, et al. 2017). While the study does an excellent job examining the influence of speed on calculation of road roughness index (and showing we will need to take that into account with the research project) and showing that crowd sourcing data is possible and effective, it did nothing regarding anomaly detection or machine-learning.

There has also been and application developed called Crater that can identify potholes and speed bumps using machine learning (although it does not do any other types of road anomalies) (Kalim, Faria, et al. 2016). Additionally, it improves location accuracy, from the built in GPS on the phone, by using open-source Google Location APIs, and also simplifies the re-orientation of the given device by using a rotation matrix that is provided by Android. A

substantial difference from their methodology is that we developed for IOS as well because it will be able to have information from a wider user base in the future.

There have also been tests on multiple vehicles and shown that at low speeds there is a sizeable difference of accelerometer data from car to car, and at higher speeds the data looks very similar (Forslöf and Jones 2015). This begs the question if a machine-learning algorithm would be more effective than simply thresholding at being able to identify road anomalies based on these discrepancies.

There has been some work done in the machine learning field, one project used a smartphone based system with a machine learning algorithm to attempt to detect 3 classifications of road anomalies and came up with an overall accuracy of 90% (Seraj et al. 2016). While they did prove the concept that a machine-learning algorithm could be used for relatively good anomaly detection, they did not collect enough training data to do more classifications, which would be substantially more useful in determining prioritization for road maintenance entities.

Most of the studies had phones in a static place in the vehicle, for example we used the center of the windshield. Since most people who drive do not put their phone statically, and there has been little research done when comparing the vibrations when the phones are at different angles and in different places in a vehicle.

CHAPTER II

METHODS

Data Collection

The first thing that had to be done was to collect accelerometer data from the phones. For the initial round of data collection the vehicle we used was a 2017 Ford Focus. For data collection we developed an iOS application that recorded GPS position, speed, acceleration in the X, Y, and Z directions, rotation about the X, Y, and Z directions, and the Unix time at each individual data point. We used and iPhone 6 to record the data with a sampling rate of 100 Hz. One of the concerns we had was that the sampling rate of the phone was not high enough to collect accelerometer data from every pothole. In order to verify that the sampling rate was high enough, we used a Raspberry PI with a sampling rate of approximately 4000 Hz to verify the data from the iPhone. Figure 1 below shows the experimental setup of the iPhone and Raspberry PI inside of the vehicle to collect vibration data. The red and green wires are attached to the Raspberry PI which is affixed to the back of the iPhone. We picked the center of the windshield to affix the iPhone to because it provided consistency in location while we collected data and more consistent vibration to train the algorithms with.

Figure 1. Image of the inside setup of the vehicle.

Machine learning algorithms require training data, which needs to be validated using a ground truth. Much of the time vibration data on regular road can look very similar to data from potholes. To address this issue, a DJI OSMO video camera was affixed to the hood of the car in order to take video of the road as we were driving. The OSMO did not had an internal clock that was accurate enough for us to match to the UNIX timestamps of the data being collected by the iPhone, so we affixed another iPhone to the front of the car in the viewing area of the OSMO with a running UNIX clock in order to be able to synchronize the ground truth to the timestamps on the accelerometer data we collected. Figure 2 shows the outside setup of the vehicle.

Figure 2. Image of the outside setup on the vehicle.

We then drove in and around the cities of Bryan and College Station in Texas to look for and drive over potholes.

Data Preprocessing

After collecting the data, we did some preprocessing. An extra piece of data preprocessing that needs to be done is to reorient the data using quaternions. This needs to be done because the machine learning algorithms assume that all the data collected has to be in the same orientation in order for accelerations to match when driving over anomalies. Reorientation will allow for the phone to start out in any position and still allow for the signals to be compared. After the data was reoriented, the data was then filtered, windowed into 100 data point windows, and labeled using the video data. In literature we found 68 different features that could be extracted to use as inputs to classify the data. Since we only had approximately 1000 windows of data, we decided to rank the features and then only use the top 10 features to classify the data. The rankings are as follows in Table 1.

Table 1. Feature Ranking

Rank	Feature	Axis
1	PSD_AvgBandPower_5to15	Z
2	MeanMedianDiff_EnvUpper_LPwindow	X
3	PSD_AvgBandPower_30to40	Z
4	PSD_RMSBandValue_35to45	Y
5	PSD_MaxBandValue_30to40	Z
6	PSD_AvgBandPower_35to45	Y
7	PSD_MaxBandValue_5to15	Z
8	PSD_RMSBandValue_0to10	Y
9	PSD_AvgBandPower_5to15	X
10	PSD_AvgBandPower_40to50	X

Algorithm Development

Since the algorithm development is not the primary purpose of this paper, this section will be relatively brief. There were three machine-learning algorithms that were used to attempt classification between the categories of smooth road, potholes, and cracks.

The first algorithm type that was attempted was a Support Vector Machine Algorithm. Using the 68 features identified earlier, the algorithm categorizes features based on training data provided and creates clusters. This in turn builds an overall model for what features correspond to which category. It then places new examples (not training data) into categories based on the features established and how well the new example fits into a specific category.

The second algorithm type used was K Nearest Neighbors algorithm. This algorithm type stores training data and cases, then compares the distance of new data from its neighbors to determine the classification. We used a K value of 5, meaning that each is compared to the class of its 5 nearest neighbors and determined based on that.

The third algorithm type used was a Binary Decision Tree algorithm. This algorithm uses features to be splits in a tree structure that eventually end at a classification. The model is created using training data, and based on how the experimental data matches up to the branches, it is classified as anomalous or not anomalous.

Different Signal Collection and Comparison

The crowd sourcing aspect of the project is one of the most interesting pieces. Thousands of people with smartphones will be able to allow for real time identification of potholes and other road anomalies. One of the things that will not be consistent though, is the position in which they affix phones. In order to try and simulate this to look at the signals, we used a 2017 Ford Focus collect data in. We also put phones 3 places in each vehicle while collecting data, on the windshield, in the pocket of the driver, and in the center console. One of the challenges that was run into was that the signals are hard to compare when the data is not reoriented to account for the fact that there is no consistent 90 degree angle. To account for this a reorientation algorithm that is described by Tundo et al. 2013 was used to reorient the data and allow for comparison to be done. The algorithm essentially works by looking at the initial angles of the phone accelerometer, and then using algebra to set the initial positions and look at all others relative to the first. Another issue is that the sampling rate is not consistent between phones so in comparing signals one of the signals needs to be down sampled. Figure 4, shown below, proves that the reorientation program was functioning properly. Since the phone was lying flat on its back in the center console, the Y-Axis accelerometer data and the Z-Axis accelerometer data were switched, which reorients the phone with the gravity vector in the Y direction (which has a value of -1).

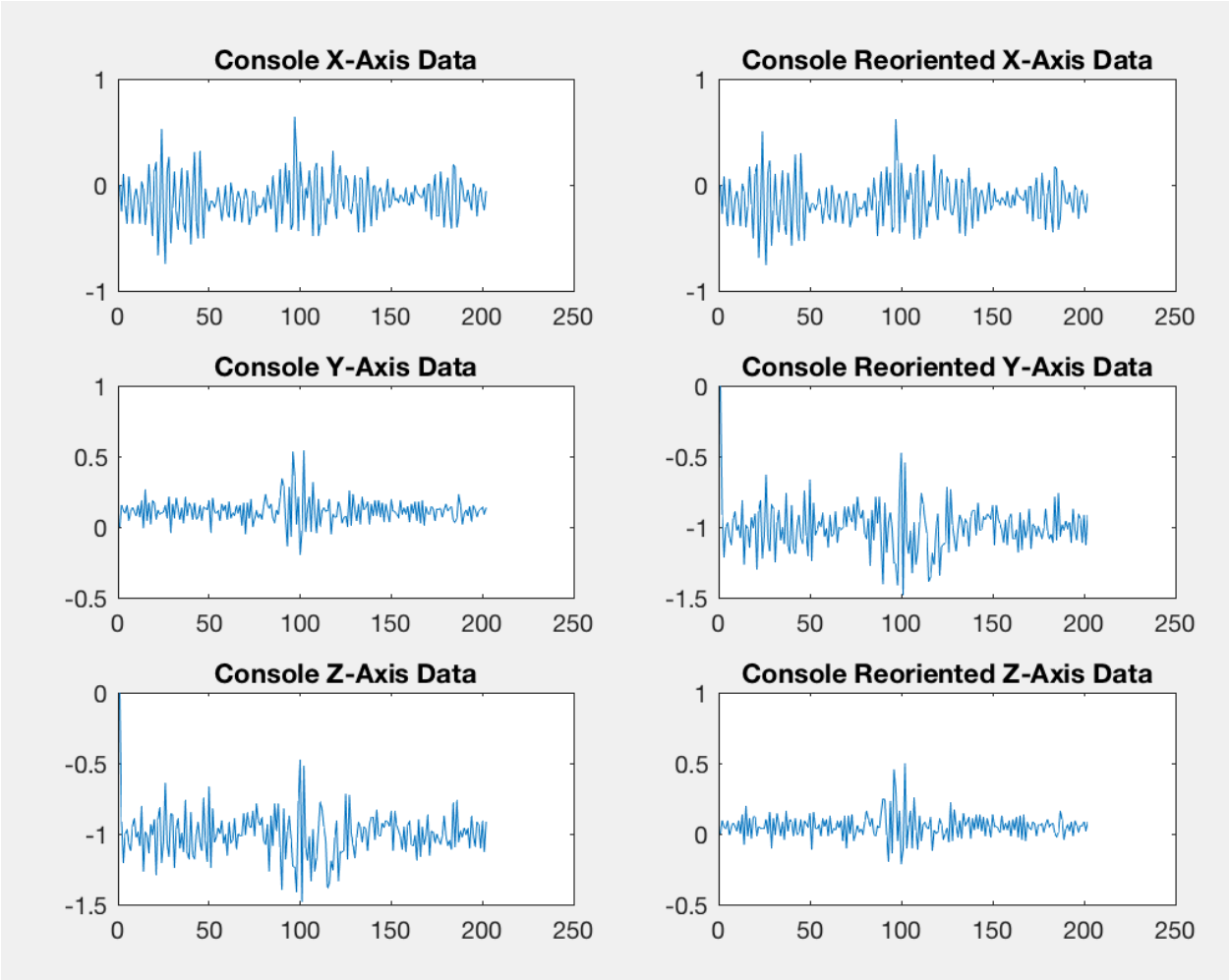

Figure 3. Console Phone Accelerometer X, Y, and Z Axis Reorientation

In comparing the signals the primary tool that was used was the Pearson cross correlation coefficient to look at signal similarity. The Pearson coefficient looks at linear correlation between Windowed segments were looked at in order to find similarities when anomalies were run over. The comparison was done between individual signals from phones placed in the same car. Another measure of comparison was looking at similarities between features from the different windows that were inputted to the machine learning algorithm to see if there were similarities between signals that could not be seen from simple cross correlation, and potentially retrain the machine learning algorithm to have similar accuracy between the sets of data.

CHAPTER III

RESULTS

Algorithm Results with New Data

There were a few problems during data collection, the first run of data incorrectly collected for both the IPhone in the pocket, and the IPhone attached to the windshield. In the second run for data, a total of 26 anomalies were detected, which was not enough to retrain the machine learning algorithm, but enough to see if the original training allowed for correct categorization of the anomalies. It was expected that overall the algorithm would be relatively accurate for the windshield data because the algorithms were trained using windshield data and that the pocket and console data would be relatively worse. Another thing to note is that NaN in all tables refer to not a number.

Table 2. Machine Learning Results with Windshield Data Input

Algorithm	**Precision**			**Recall**			**F_1 Score**		
	Cracks	Potholes	Smooth	Cracks	Potholes	Smooth	Cracks	Potholes	Smooth
Dec-Tree	0.16	1.00	1.00	1.00	0.39	0.38	0.27	0.56	0.55
kNN	0.00	0.00	1.00	NaN	0.000	0.11	NaN	NaN	0.19
SVM	0.89	0.86	1.00	0.94	0.86	0.75	0.92	0.86	0.86

The results for the data shown in Table 2, indicate that the SVM performed the best across all performance metrics. Since the majority of the anomalies inputted were cracks, which was different from the data that the algorithms were trained with, it was surprising that the SVM actually performed better classifying cracks than it has previously.

Table 3. Windshield Data Accuracy

Windshield Classification Accuracy	
Decision Tree	0.45
kNN	0.10
SVM	0.90

As shown in Table 3, the accuracy for classification of anomalies was on par with the results obtained for SVM with the original set of classification data. It is unknown as to the reasons for kNN and Decision Tree Algorithms working poorly for potholes, but it may have to do with the fact that the number of potholes was significantly lower than the number of cracks in the testing data.

The next set of data that was inputted in the machine learning algorithms was the center console data. This data did not perform as well as the windshield data, although as you can see in Table 5, the SVM and Decision Tree overall accuracy were approximately fifty percent. In Table 4, you can see that the Precision, Recall, and F1 scores were all around the same for cracks and potholes with the SVM. This result is likely due to the fact that the vibrations are directly coming from the vehicle, although since it is in a different location the vibrations will be somewhat different.

Table 4. Machine Learning Results with Console Data Input

Algorithm	**Precision**			**Recall**			**F_1 Score**		
	Cracks	Potholes	Smooth	Cracks	Potholes	Smooth	Cracks	Potholes	Smooth
Dec-Tree	0.21	1.00	1.00	1.00	0.37	0.50	0.35	0.54	0.67
kNN	0.05	0.00	1.00	0.20	0.00	0.14	0.08	NaN	0.25
SVM	0.53	0.29	1.00	0.67	0.25	0.50	0.59	0.27	0.67

Table 5. Console Classification Accuracy

Console Classification Accuracy	
Decision Tree	0.48
kNN	0.14
SVM	0.52

The pocket data performed terribly. The Precision, Recall, and F1 score in Table 6 were all sub 20 percent. The classification accuracy in Table 7 was even smaller than in all of the other iterations. A possibility to consider is the damping effect that the pocket has on the phone.

The phones vibrations are significantly damped because it is between two pieces of fabric and there is a significant amount of extra friction. The vibrations might also vary depending on the tightness of the pockets and type of material.

Table 6. Machine Learning Results with Pocket Data Input

Algorithm	**Precision**			**Recall**			**F_1 Score**		
	Cracks	Potholes	Smooth	Cracks	Potholes	Smooth	Cracks	Potholes	Smooth
Dec-Tree	0.00	0.00	1.00	NaN	0.00	0.12	NaN	NaN	0.21
kNN	0.00	0.00	1.00	NaN	0.00	0.11	NaN	NaN	0.20
SVM	0.11	0.00	1.00	1.00	0.00	0.12	0.19	NaN	0.21

Table 7. Pocket Classification Accuracy

Pocket Classification Accuracy	
Decision Tree	0.10
kNN	0.10
SVM	0.17

Cross Correlation of Windows in Time Domain

While the results for the windshield data confirmed that the SVM was trained well and had a similar accuracy, the console phone data and pocket phone data performed significantly worse when plugged into the algorithms. One thing to examine in looking for reasons as to why they did not perform as well together, is the similarity between signals, using the Pearson cross correlation coefficient. There were 26 windows that were each 100 data points long in each axis direction. The cross correlation coefficient of each was taken and then averaged. As Table 8 shows below, there is no discernable correlation between any of the signal sets. While this helps explain one of the reasons why classification was not optimal for the pocket and console, another thing to look at is the correlation between the features to see if using other features to train the machine-learning algorithm would improve the performance of the machine-learning algorithm.

Table 8. Pearson Cross Correlation Coefficient

	Crack Windshield and Console	Crack Windshield and Pocket	Crack Console and Pocket	Pothole Windshield and Console	Pothole Windshield and Pocket	Pothole Console and Pocket
X-Axis	-0.03	0.02	0.03	0.03	-0.15	0.04
Y-Axis	-0.00	0.06	0.08	-0.03	0.18	0.07
Z-Axis	-0.05	-0.02	-0.02	0.04	0.02	0.02

Cross Correlation of Features

Since there was no discernable correlation between the windows of signals in their raw form, it was decided it would be advantageous to look for trends of correlation within the features that were extracted and train the machine-learning algorithm in an attempt to classify each set of data with a similar accuracy. Table 9 shows a list of the features extracted that were used in comparison. There were multiple bins that were used for the last three features.

Table 9. List of Features Extracted

Extracted Features Used in Signal Comparison
Window Maximum
Window Minimum
Peak to Peak
Root Mean Square Value
Variance
Standard Deviation
Mean Peak
Maximum Peak
Root Mean Square Peak
Minimum Signal Trough
Root Mean Square Trough
Peak to Peak Envelope
Envelope Mean
Envelope Mean and Median Difference
Signal Periodogram
Average Signal Power
Average Power of Power Spectral Density Bins
Root Mean Square of Bins
Maximum of Bins

When extracting the features, to optimize the current algorithm (which is trained using only windshield data) doing the cross correlation of features in relation to the windshield data is important because high correlations with the windshield will be more likely to be classified correctly. When comparing the features described in Table 1 in the Methods section to Table 10, one of the main differences is that there are essentially no power spectral density features. The features which produced the highest correlation when comparing the datasets, were all time domain features. This makes sense because the frequency components are going to be different because the vibration patterns differ depending on the location in the vehicle that the phone resides.

Table 10. Feature Pearson Cross Correlation Coefficient

Feature	Correlation Between Windshield and Pocket	Correlation Between Windshield and Console
Maximum Z-Axis	.71	.67
Minimum Y-Axis	.70	.69
Peak to Peak Y-Axis	.92	.77
RMS X-Axis	.89	.65
RMS Z-Axis	.78	.68
Standard Deviation Z-Axis	.71	.67
Mean Peak Z-Axis	.78	.74
Max Peak Y-Axis	.89	.64
Mean Trough X-Axis	.89	.63
Envelope Peak to Peak X-Axis	.83	.91
Envelope Peak to Peak Z-Axis	.78	.89
Difference of Envelope Mean and Median Z-Axis	.65	.65

Algorithm Results with New Features Chosen

Using the 12 features identified in order, the algorithm was retrained using the same windshield data that was collected before the pocket and console data were collected. As seen

below in Tables 11 and 12, the results were significantly worse for the windshield data with approximately a 45 percent accuracy rating.

Table 11. Machine Learning Results with Windshield Data Input

Algorithm	**Precision**			**Recall**			**F_1 Score**		
	Cracks	Potholes	Smooth	Cracks	Potholes	Smooth	Cracks	Potholes	Smooth
Dec-Tree	0.00	0.00	1.00	NaN	0.30	0.75	NaN	0.47	0.86
kNN	0.00	.29	1.00	NaN	0.33	0.31	NaN	0.31	0.47
SVM	0.00	0.00	1.00	NaN	0.29	0.82	NaN	0.45	0.90

Table 12. Windshield Classification Accuracy

Pocket Classification Accuracy	
Decision Tree	0.46
kNN	0.31
SVM	0.46

The console data, collected in Tables 13 and 14 had almost the exact same accuracy as seen for the windshield data, although this is a decrease of around 10 percent from the previously trained algorithm.

Table 13. Machine Learning Results with Console Data Input

Algorithm	**Precision**			**Recall**			**F_1 Score**		
	Cracks	Potholes	Smooth	Cracks	Potholes	Smooth	Cracks	Potholes	Smooth
Dec-Tree	0.00	1.00	1.00	NaN	0.35	0.60	NaN	0.52	0.75
kNN	0.00	0.43	1.00	NaN	0.43	0.32	NaN	0.43	0.49
SVM	0.00	1.00	1.00	NaN	0.30	0.75	NaN	0.47	0.86

Table 14. Console Classification Accuracy

Pocket Classification Accuracy	
Decision Tree	0.46
kNN	0.34
SVM	0.46

The pocket data was the most surprising though, with the accuracy increasing 30 percent with the SVM algorithm, and the other algorithms having a marked accuracy increase as well.

As seen in Tables 15 and 16, while still slightly performing slightly worse than the other algorithms, the results for the pocket data were much more consistent.

Table 15. Machine Learning Results with Pocket Data Input

Algorithm	**Precision**			**Recall**			**F_1 Score**		
	Cracks	Potholes	Smooth	Cracks	Potholes	Smooth	Cracks	Potholes	Smooth
Dec-Tree	0.000	0.14	1.0000000	NaN	0.14	0.32	NaN	0.14	0.49
kNN	0.000	0.00	1.000	NaN	0.00	0.26	NaN	NaN	0.42
SVM	0.000	1.00	1.0000000	NaN	0.27	1.00	NaN	0.42	1.00

Table 16. Pocket Classification Accuracy

Pocket Classification Accuracy	
Decision Tree	0.29
kNN	0.26
SVM	0.46

CHAPTER IV

CONCLUSION

In this work, we have tested three smart phone positions for collecting vibration data for detecting road abnormality: on windshield, on console and in pocket. The similarity of the raw signals were assessed using Pearson coefficients, and data quality was assessed using road classification accuracies through the machine learning algorithms. Since the results were so poor, we tried training the machine learning algorithms with features that had a high Pearson coefficient between the data sets. The results showed that while there was a decrease in the accuracy in categorizing the windshield data and console data, there was an increase in the accuracy in categorizing the pocket data. Although using the new features contributed to more consistent results, they were overall worse and inaccurate in classifying anomalies. One lasting question is, what features not only provide similar results for sets of data that have been reoriented, but also are precise in classification.

Unfortunately there was not enough time to collect and label enough data from three different phones to train the machine learning algorithms and see the results. Ultimately though, a study that looked at large amounts of data from different orientations inputted into training a machine learning algorithm for road anomaly detection would be a direction to go.

One of the things that would be good to look and was overlooked by this paper is road roughness index. While there is not a huge demand for determining potholes and road anomalies, the things that could be done with an accurate roughness index are numerous. From using roughness as a factor into GPS routing, to looking at long-term suspension performance the possibilities are endless.

REFERENCES

Alessandroni, Giacomo, et al. "A Study on the Influence of Speed on Road Roughness Sensing: The SmartRoadSense Case." *Sensors*, vol. 17, no. 2, July 2017, p. 305., doi:10.3390/s17020305.

Cao, Yuejian, et al. "Evaluation of Pavement System Based on Ground-Penetrating Radar Full-Waveform Simulation." *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2227, 2011, pp. 71–78., doi:10.3141/2227-08.

Chen, Syuan-Yi, et al. "Road Conditions Detection Using Arduino Based Sensing Module and Smartphone." *2015 IEEE International Conference on Consumer Electronics - Taiwan*, 2015, doi:10.1109/icce-tw.2015.7216884.

Forslöf, Lars, and Hans Jones. "Roadroid: Continuous Road Condition Monitoring with Smart Phones." *Journal of Civil Engineering and Architecture*, vol. 9, no. 4, 2015, doi:10.17265/1934-7359/2015.04.012.

Kalim, Faria, et al. "CRATER: A Crowd Sensing Application to Estimate Road Conditions." *IEEE Access*, vol. 4, 2016, pp. 8317–8326., doi:10.1109/access.2016.2607719.

Kalra, Nidhi, et al. "Analyzing Driving and Road Events via Smartphone." *International Journal of Computer Applications*, vol. 98, no. 12, 2014, pp. 5–9., doi:10.5120/17233-7561.

Seraj, Fatjon, et al. "RoADS: A Road Pavement Monitoring System for Anomaly Detection Using Smart Phones." *Lecture Notes in Computer Science Big Data Analytics in the Social and Ubiquitous Context*, 2016, pp. 128–146., doi:10.1007/978-3-319-29009-6_7.

Tundo, M. D., et al. "Correcting Smartphone Orientation for Accelerometer-Based Analysis." *2013 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2013, doi:10.1109/memea.2013.6549706.