



CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies, CENTERIS / ProjMAN / HCist 2017, 8-10 November 2017, Barcelona, Spain

Anomaly Detection in Roads with a Data Mining Approach

Nuno Silva^{a*}, João Soares^a, Vaibhav Shah^a, Maribel Yasmina Santos^a, Helena Rodrigues^a

^aALGORITMI Reasearch Centre, Universidade Do Minho, Azurém Campus, 4804-5333 Guimarães Portugal

Abstract

Road condition has an important role in our daily live. Anomalies in road surface can cause accidents, mechanical failure, stress and discomfort in drivers and passengers. Governments spend millions each year in roads maintenance for maintaining roads in good condition. But extensive maintenance work can lead to traffic jams, causing frustration in road users. In way to avoid problems caused by road anomalies, we propose a system that can detect road anomalies using smartphone sensors. The approach is based in data-mining algorithms to mitigate the problem of hardware diversity. In this work we used scikit-learn, a python module, and Weka, as tools for data-mining. All cleaning data process was made using python language. The final results show that it is possible detect road anomalies using only a smartphone.

© 2017 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of the scientific committee of the CENTERIS - International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies.

Keywords: Road Anomalies, Data Mining, Data Analytics

1. Introduction

Road condition has an important role in our daily live. Anomalies in road pavement can cause vehicle mechanical failure or even accidents. Also, roads in bad conditions can affect driver and passenger comfort and increase stress.

* Corresponding author. Tel.: +351 967161737.

E-mail address: b7204@algoritmi.uminho.pt

Finally, governments spend millions of euros in road maintenance, which can cause traffic jams, increase driver stress, fuel consumption and cause delays in our daily live.

A road condition/anomaly tracking system can help governments planning and scheduling road maintenance in a way that decreases the impact in people daily live, while, an anomaly warning system for drivers can increase anomaly awareness, which may decrease the number accidents, costs in car maintenance and driver stress. Also, when planning a trip, information about roads condition can increase comfort both in driver and passengers^{1,2}.

However, detecting and surveying road condition/anomalies requires expensive and specially designed equipment and vehicles that cost considerable amounts of money, while also requiring specialized workers to operate them³. On the other hand, most drivers have smartphones, which are equipped with several different sensing capabilities, namely inertial and location sensors⁴. Also, smartphones are part of our everyday lives, with most people owning, at least, one device. Thus, these devices could be used as road monitoring equipment, “transforming” every driver/car into specialized road-monitoring technicians/vehicles.

In this work, we describe a system that, based on data collected by smartphone sensors, automatically detects and classifies road anomalies using data mining algorithms. Specifically, we describe the methodology used for the data mining process, and discuss some problems and challenges resulting from this process.

However, the use of smartphone devices imposes several problems and limitations when compared to the more expensive and specialized road-monitoring equipment. Specifically, smartphone devices use less expensive sensors that typically offer less sensitivity and precision when compared to specialized equipment. This is especially true for inertial and GPS sensors. Additionally, the huge variety of smartphone and sensor makes and models result in a data heterogeneity problems, since different sensors typically measure different values for the same physical action³.

This heterogeneity is not only related to smartphones and sensors. The automotive park suffers from the same problems, with many different car makes and models, distinct suspension types and tires. All this has direct influence and impact on the collected data, thus existing solutions that detect and categorize road anomalies based on pre-defined thresholds are not suited for real world scenarios^{5,6,7}. Additionally, GPS sensors of smartphones typically have an accuracy error ranging from a couple to tens of meters^{8,9}, meaning that the same physical location can have distinct coordinates (latitude and longitude) in different acquisitions. Furthermore, driving speed has a considerable impact in vehicle vibration, creating a higher vibration amplitude in accelerometer data^{5,10}.

2. Related Work

When we talk about road condition monitoring systems, we typically talk about data centered systems. Data is typically collected by devices and analyzed to better understand the problem in hand, and to acquire knowledge about it. However, raw data is difficult to analyze and hard to understand. Data needs to be transformed in order to be more comprehensive, increasing its usability in data mining algorithms.

In literature, we see many different approaches to solve these problems. In the data acquisition stage, labeling is one of the hardest to solve. Seraj et al.¹¹ uses video cameras to analyze road condition and a subtitle editor to annotate anomalies. Speed is another problem to take into consideration, as GPS signals can be erratic. Also, routes need to be well defined as to avoid high traffic and pedestrian volumes, not only due to safety reasons, but also to prevent data quality from being compromised by these interferences. Finally, diversity of vehicles, drivers and recording/sensing devices are important measures that need to be taken for preventing algorithm overfitting.

In the data cleaning and transformation stage, one needs to clean and process erratic data, and know which features must be extracted from data. When collecting data from accelerometer sensors, a considerable amount of noise is collected due to events like, slamming doors or vibration caused by motor running. To remove this noise Vittorio et al.¹ and Mukherjee et al.¹² use a low pass filter. Douangphachanh et al.¹³ use a high pass filter, based on the Android Developer Reference, to remove gravity from all accelerometer axis, combined with a low pass filter. For erratic data, Chen et al.¹⁴ deals with it by deleting it or using interpolation to fill that gaps. For extracting features, Nomura et al.⁴ calculates vertical acceleration variances, while Mukherjee et al.¹² uses acceleration means and Bello-Salau et al.¹⁵ uses standard deviation of accelerometer data, and presents a new measure called “z-difference square”, that uses the square of z-axis magnitude($Z_0 - Z_1$) in order to get only positive values. Rajamohan et al.¹⁶ use standard deviation of all three accelerometer axis. Others use more complex algorithms to do signal analysis. Seraj et al.¹¹ uses Stationary Wavelet Transformation (SWT) at four levels of decomposition to analyze data from gyroscope sensors Joubert et al.¹⁷ use Kalman filters, and Mukherjee et al.¹² use Tturn, Tmin, Tbank and

Twin filters to help reducing the number of false negative, while Hanson et al.¹⁸, Chug et al.¹⁹ uses Discrete Fourier Transform (DFT). For detecting events and classification, Rajamohan et al.¹⁶ uses K-Nearest Neighbour (K-NN) algorithm, while Chen et al.⁵ uses Gaussian Mixture Model (GMM) and Improved Gaussian Mixture Model (i-GMM). Additionally, algorithm success rate can be improved by using filters to eliminate some types of events. For example, Chen⁵ presents filters with specific goals, like remove events that happened at zero velocity (eg., slam doors).

3. Proposed Approach for Detecting Road Anomalies

For dealing with the problems described previously, we propose a data mining approach. As data mining algorithms are able to find patterns in data, this can mitigate problems of hardware diversity and speed influence, for example. At the end, we expect to have better results than threshold based approaches, where these problems have a strong impact.

For implementing this data mining approach, we will use CRISP-DM²⁰ framework to guide us in all process. CRISP-DM is divide in six phases. Each phase is divided in small tasks in order to deal with a smaller set of problems. At the end of each phase, depending of the generated outcome, we execute another phase or task, as described next.

3.1. Business Understanding

When studying the problem domain, we observe that the most common anomalies are potholes, manholes and bumps²¹. These three categories can categorize most of the anomalies found in our daily life. Besides that, we add one more category – Others: where we join all anomalies that do not fit in the previous three categories.

Thus, we have:

- Potholes – are mainly caused by bad pavement quality or below surface problems;
- Manholes – are holes that allow workers access to some underground location. Normally allow access to sewers and underground city infrastructures;
- Bumps (Long and Short) – normally man made, and usually used to slow down vehicles in crosswalk proximity;
- Others – anomalies that do not fit in any of the above described categories. These are anomalies without any specific format. These can be depressions or asphalt elevations.

For this project, and as a result of road conditions in our geographical area (Braga-Portugal), we decided to focus only in four anomalies types: *manholes*, *long* and *short bumps*, and *others*, leaving behind potholes. The reason for this is that there are few potholes in our geographical area, so besides being difficult to acquire representative data from them, they do not represent typical anomalies. Also, we decided to divide bump types in two different categories: *long* and *short bumps*. This division results from the possible danger level of each type. Long bumps are more dangerous then short bumps, also, there are an ample number of long and short bumps in our geographical area.

3.2. Data Understanding

As we described above, we will try to detect anomalies using smartphones as data acquisition devices. We could use only 3-axis accelerometer to detect anomalies, but we need GPS coordinates to determine anomalies location. Speed (extracted from GPS sensor) is relevant due to its influence in accelerometer data magnitude. For example, a high-speed causes high values in accelerometer data, which in turn can have a correlation with the detection of anomalies. We also collect timestamps to maintain anomaly history. This information could be used, for example, to plot anomalies in a map, showing the evolution of the road conditions. In this way, we can create a service to be used for drivers and government entities.

At this moment, all collected data (accelerometer, latitude, longitude, speed, timestamp, anomaly) are collected by smartphone as we can observe in Fig. 1. The value ranges of each attribute are:

- Timestamp: positive and integer values
- Latitude and longitude: negative or positive values with decimal places (float)
- Speed: positive values with decimal places in meters per second (float)
- Accelerometer <x, y, z>: positive or negative values with decimal places (float)

- Anomaly: string with the anomaly name or blank attribute.

All attributes may have null data due to sensor errors; for example, data from GPS can be null due to no GPS connection.

Timestamp	GPS.latitude	GPS.longitude	GPS.speed	Linear Acceleration.x	Linear Acceleration.y	Linear Acceleration.z	Anomaly
1483997963986	41.562309	-8.3948427	6.75	-1.5005134	-1.8224201	-2.0122588	
1483997964004	41.562309	-8.3948427	6.75	-1.2993184	0.45824718	0.065624	
1483997964039	41.562309	-8.3948427	6.75	-2.232779	-0.87073517	-2.8772116	
1483997964058	41.562309	-8.3948427	6.75	-1.5570658	-1.2681322	-0.62367636	
1483997964078	41.562309	-8.3948427	6.75	0.18805826	1.2973471	1.3578396	

Fig. 1. Collected data sample

For acquiring data, we developed an Android application that allow us acquire data from the sensors above described. Besides that, we choose 50Hz sample rate, because, assuming 30cm as anomaly size, for a 50Km/h velocity (maximum speed allowed by law in city) we need a little less than 50Hz. In this application, we can label anomalies simultaneously with data acquisition from smartphone sensors. As any touch in a smartphone could change accelerometer values and infringe the law, we used headphones with buttons, where each button corresponds to one anomaly type. In this way, we avoid touching the smartphone. Labelling anomalies allows us to use supervised learning algorithms in the modelling phase, as described next.

All data is stored in *.csv* files. These *.csv* files is uploaded when the smartphone has Internet connection.

The data acquisition process was made with 3 different cars and 2 different smartphones, in distinct days. To avoid external interferences, most data acquisition occurred during the night and in low traffic zones. Thus, the process is safest and we avoid other problems like traffic jams and crossing roads pedestrians, which would compromise data quality.

As we demonstrate in Table 1, we finish our data acquisition phase with 157 anomalies recorded, divided into four categories, corresponding to a total of 143 files, and 10.3Mb.

Table 1. Collected Anomalies Quantity and Type

	Manhole	Short Bump	Long Bump	Other	Total
Anomalies Number	36	53	51	17	157

3.3. Data Preparation

After data acquisition, we started the data preparation phase. In this step, we need to clean data, i.e. treat *null* values, deal with possible inconsistent data (negative values in timestamp and too high speed can point to some sensors problems), extract features, transform data (using normalization technique for example), as we present in Fig. 2.

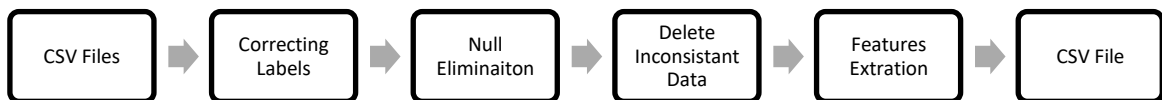


Fig. 2. Data Preparation Process

The labeling data process was performed simultaneously with the driving activity. In this specific case, as we use a 50hz sample rate in the accelerometer sensors, it was impossible to label the exact corresponding data. So, it was necessary to review all data and correct the labeling. This was a manual process, using graphical visualization (Fig. 3) to see where the anomaly started and ended.

With this information, it was possible to fill the anomalies labels in all data. As can be seen in Fig. 4, the anomaly label was introduced, changing the moment when the anomaly started and ended. This process was necessary to grant a solid ground-truth for training the data mining algorithms.

After establishing a solid ground-truth, we used a Python script and some Python libraries (e.g., Pandas and numPy) for the cleaning and features extraction process. In this process, *null* data was eliminated. Rows with speed values of 0 were deleted, since no anomalies can be detected when the vehicle is stationary.

The feature extraction process started with the use of a sliding window technique with an overlap factor. The use of this strategy is due to the fact that, for this specific case, one event, i.e., road anomaly, is represented by more than one data row. Thus, the number of rows that represent an anomaly are unknown ‘a priori’, with some anomalies represented by 35 rows, and others by 90. This is related to the anomaly physical size, and vehicle speed.

From this sliding window, we calculate the average of the existing attributes, and calculate new attributes, such as 3 axis accelerometer max, min and mean; the integral value using approximation by Simpson’s rule, among others, thus, transforming a set of rows into a single row, minimizing the loss of relevant information.

However, as a result of using a sliding window strategy with overlap, one physical anomaly can be included in two or more windows (depending of the anomaly and window size) (Fig. 5). We refer to this as “ghost anomalies”. As result, in the final file, one physical anomaly can be represented by more than one row.

We use a sliding window of 125 samples with 60% of overlap. In this context, it is considered that one road anomaly exists when: $(rows\ in\ window\ with\ anomaly) > (25\% * rows\ in\ window\ without\ anomaly)$.

At the end of this phase, we finished with only one .csv file and 39 attributes and 1311 rows, when starting from 77 .csv files, 8 attributes and 63819 rows.

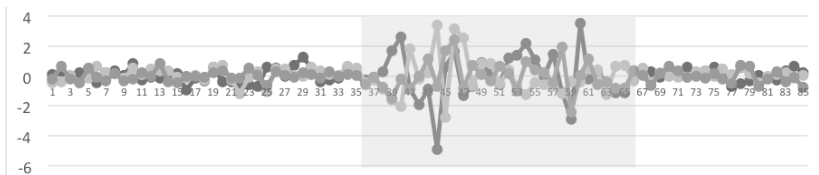


Fig. 3. Linear Acceleration example with anomaly

a)		b)					
415606	-8.334	7.5	-0.8884	0.21258	1.14566		
415606	-8.334	7.5	-4.8318	3.40438	-0.715		
415606	-8.334	7.5	0.5782	-2.7966	1.67689		
415606	-8.334	7.5	1.61975	3.16235	2.42455		
415606	-8.334	7.5	-1.2818	2.53363	-0.8265	Unlabeled Manhole	
415606	-8.334	7.5	-0.7138	-0.6574	0.74407	Unlabeled Manhole	
415606	-8.334	7.5	0.31241	0.84664	0.03836	Unlabeled Manhole	
415606	-8.334	7.5	0.10088	0.82568	-0.2316	Unlabeled Manhole	
415606	-8.334	7.5	-0.5344	-0.5046	0.63546	Unlabeled Manhole	
415606	-8.334	7.5	1.20864	0.40887	0.14393	Unlabeled Manhole	
415606	-8.334	7.5	1.36343	-0.7065	-1.0218	Unlabeled Manhole	
415606	-8.334	7.5	2.17555	-1.2522	0.3353	Unlabeled Manhole	
415606	-8.334	7.5	1.09527	-0.4531	0.41411	Unlabeled Manhole	
415606	-8.334	7.5	-0.0365	-0.532	0.21503	Unlabeled Manhole	
415606	-8.334	7.5	1.44593	-0.7163	-0.4234	Unlabeled Manhole	
415606	-8.334	7.5	-1.1039	-1.1912	1.36937	Unlabeled Manhole	
415606	-8.334	7.5	-2.9065	-0.3724	-2.4235	Unlabeled Manhole	
415606	-8.334	7.5	3.60885	-0.3903	0.07119	Unlabeled Manhole	

Fig. 4. a) Anomaly label before manual filling b) Anomaly label after the manual filling

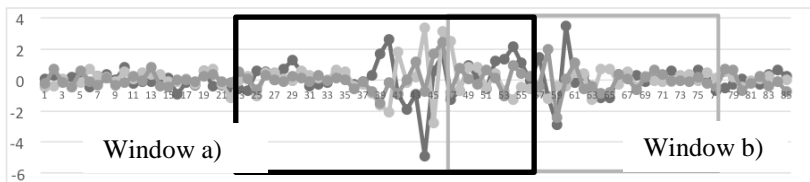


Fig. 5. Linear Acceleration example with anomaly and two windows with 50% overlap

3.4. Modeling

Analyzing our problem and our goals, considering that we have a solid ground-truth with anomalies correctly labeled, we face a classification problem. For this we can use supervised learning algorithms, such as decision trees, for example. This way, in order to choose the best algorithm for this problem and study possible alternatives, we perform the following tasks:

1. Run classification algorithms with all dataset attributes;
2. Analyze the results and choose the best models;
3. Run classification algorithms with the most important attributes, identified by attribute selection models;
4. Analyze results and choose the best models;

First, we use the algorithms with the standard pre-defined values. After analyzing all the results, some changes can be made in each algorithm parameters looking for better results, bearing in mind that tweaking the algorithm can cause model overfitting.

At this moment, we anticipate that tree-based models (like Random Forest (RF), Gradient Boosting (GB) or Decisions Trees (DT)) can accomplish good results for this problem. Neural Networks (NN) can also accomplish good results, but can have performance problems in training, depending of data size. Support Vector Machines (SVM) have the same training performance problems as NN.

3.5. Evaluation

In this section, we will analyze the obtained results. We evaluate the model capability of detecting anomalies and types. For that, the metrics that will be used are: score for general overview and confusion matrix for a more detailed analysis.

3.5.1. First Experience – Full attributes

As we can observe in Table 2, tree based algorithms (GB and DT) are the ones that obtained better results. The MPL Classifier, a NN algorithm, came next in the table, with approximately the same score as the DT. Linear SVC, a SVM algorithm, had the worst score.

As expected, GB (Table 3) and DT (Table 4) show better performance in classifying all anomalies types. We can also see that the dataset presents more samples without any anomaly. This can have a big impact in anomaly detection as there is a significant difference between the number of samples in this class, when compared to the other classes. Looking to the “Other” class, we can see a worst performance in the DT algorithm, comparing with GB. This can be caused by two things: first, this is the smallest class in the dataset. Less examples to train the algorithms leads to difficulties in the classification process. Second, the way this class behaves presents some similarities with the “Unlevelled Manhole” class, reason why we see, in Table 4, eight “Other” class samples misclassify as “Unlevelled Manhole”.

Table 2. Algorithms Score with the full dataset

Algorithms	Score
Gradient Boosting	0.8705
Decision Tree	0.8071
MLP Classifier	0.7868
Gaussian NB	0.7385
Linear SVC	0.4619

Table 3. Gradient Boosting Confusion Matrix

Classified as →	No Anomaly	Unlevelled Manhole	Short Bump	Long Bump	Other	% of correct classifications
No Anomaly	206	7	4	1	4	92,8%
Unlevelled Manhole	4	18	2	1	0	72,0%
Short Bump	5	0	28	2	1	77,8%
Long Bump	7	3	3	31	0	70,5%
Other	2	2	2	1	6	46,2%

Table 4. Decision Tree Confusion Matrix

Classified as →	No Anomaly	Unlevelled Manhole	Short Bump	Long Bump	Other	% of correct classifications
No Anomaly	256	7	3	8	2	92,3%
Unlevelled Manhole	7	12	2	2	2	48,0%
Short Bump	6	2	23	5	0	63,9%
Long Bump	5	7	6	25	1	56,8%
Other	0	8	1	2	2	15,4%

3.5.2. Second Experience – Best attributes

To improve training performance and verify if some attribute is inducing noise in the model, we used Weka attribute selection algorithms to visualize the most important ones to our classification problem.

As we can observe in Fig. 6, Weka CFS Subset Evaluator, with Greedy Stepwise as search method, gives the ten most valuable attributes in our dataset, ordered from the most important to the less important ones. As stated in Weka’s documentation, Greedy Stepwise works by starting with no/all attributes or from an arbitrary point in the space, and stops when adding/deleting any remaining attribute results in a decrease in the model performance.

In Table 5, we can perceive an improvement in almost all algorithms scores. The only exception was the NN (MLP Classifier), which decrease can be explained by the NN complexity and their parameters that need to be adjusted to a new dataset. Nevertheless, the differences in this case are not significant.

```
Attribute Subset Evaluator (supervised, Class (nominal): 16 anomaly):
CFS Subset Evaluator
Including locally predictive attributes

Ranked attributes:
0.382  8  acc_y_min
0.427  6  acc_y_max
0.445  13 acc_z_min
0.454  9  acc_y_std_array
0.457  31 square_dif_y_max
0.458  38 square_dif_z_std
0.457  1  acc_x_max
0.457  12 acc_z_mean
0.455  3  acc_x_min
0.453  36 square_dif_z_max

Selected attributes: 8,6,13,9,31,38,1,12,3,36 : 10
```

Fig. 6. Weka Attribute Selection – CFS Subset Evaluator with Greedy Stepwise search method

Table 5. Algorithms scores with 10 chosen attributes by Weka

Algorithms	Score	Difference
Gradient Boosting	0.8756	+ 0.0051
Decision Tree	0.8248	+ 0.0177
MLP Classifier	0.7817	- 0.0051
Gaussian NB	0.7918	+ 0.0533
Linear SVC	0.8274	+ 0.3655

Linear SVC has the best improvement (from 46% to 82%), but the obtained model continues to fail (Table 6) in classifying the “Unlevelled Manhole” and “Other” classes.

Table 6. LinearSVC Confusion Matrix – 10 attributes dataset

Classified as →	No Anomaly	Unlevelled Manhole	Short Bump	Long Bump	Other	% of correct classifications
No Anomaly	269	1	2	1	3	97,5%
Unlevelled Manhole	14	3	1	3	4	12,0%
Short Bump	6	3	21	6	0	58,3%
Long Bump	4	5	1	33	1	75,0%
Other	7	0	4	2	0	00,0%

These results show that more attributes do not mean better results, as the number of attributes will impact in training time and can lead to overfitting, or add noise, making harder the classification process.

4. Conclusions and Future Work

The main goal of this project is detecting road anomalies using smartphone sensors and data mining algorithms. Data was labeled and divided into five classes. We extracted features, creating new attributes and using information already present, and we used a sliding window strategy with overlap to transform a set of samples into one row.

Due to classification problems, we tested five different algorithms, accomplishing a best score of 0.8705. As we have a dataset with 39 attributes, that can induce overfitting and noise, decreasing algorithms performance, the Weka tool was used for gathering the ten best attributes. We also run the same algorithms with this new dataset, and observed an improvement in almost all of them, accomplishing a best score of 0.8756.

During this work, we found that “Other” and “Unlevelled Manhole” can be misclassified. We believe that combining them together would achieve better results. Other issues came from using sliding window strategy with

overlap, since it can create more than one row for each physical anomaly (*ghost anomalies*). This problem can be solved doing post-processing, eliminating or clustering anomalies with identical GPS coordinates.

In order to obtain better results, we can use other strategies. We can leave behind the sliding window strategy and use Hidden Markov Models, try using unsupervised learning techniques, as K-means clustering to see if we can segment the anomalies or deep learning, although this requires a vast quantity of data.

Acknowledgments

This research is sponsored by the Portugal Incentive System for Research and Technological Development. Project in co-promotion nº 002797/2015 (INNOVCAR 2015-2018)

References

1. Vittorio A, Rosolino V, Teresa I, Vittoria CM, Vincenzo PG, Francesco DM. Automated Sensing System for Monitoring of Road Surface Quality by Mobile Devices. *Procedia - Soc Behav Sci*. 2014;111:242-251. doi:10.1016/j.sbspro.2014.01.057.
2. Douangphachanh V, Oneyama H. Exploring the Use of Smartphone Accelerometer and Gyroscope to Study on the Estimation of Road Surface Roughness Condition. In: *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics*. SCITEPRESS - Science and Technology Publications; 2014:783-787. doi:10.5220/0005117407830787.
3. Douangphachanh V, Oneyama H, Engineering E. A Study on the Use of Smartphones for Road Roughness Condition Estimation. *Proc East Asia Soc Transp Stud*. 2013;9(2007):14. doi:10.1186/1687-1499-2014-114.
4. Nomura T, Shiraiishi Y. A method for estimating road surface conditions with a smartphone. *Int J Informatics Soc*. 2015;7(1):29-36.
5. Chen K, Tan G, Lu M, Wu J. CRSM: a practical crowdsourcing-based road surface monitoring system. *Wirel Networks*. 2016;22(3):765-779. doi:10.1007/s11276-015-0996-y.
6. Kalra N, Chugh G, Bansal D. Analyzing Driving and Road Events via Smartphone. *Int J Comput Appl*. 2014;98(12):5-9. doi:10.5120/17233-7561.
7. Mohan P, Padmanabhan VN, Ramjee R. Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones. *Proc 6th ACM Conf Embed Netw Sens Syst - SenSys '08*. 2008:323. doi:10.1145/1460412.1460444.
8. Bo C, Jung T, Mao X, Li X, Wang Y. SmartLoc: sensing landmarks silently for smartphone-based metropolitan localization. *EURASIP J Wirel Commun Netw*. 2016;2016(1):111. doi:10.1186/s13638-016-0603-7.
9. Gps Error Analysis. In: *Global Positioning System: Theory and Applications, Volume I*. Washington DC: American Institute of Aeronautics and Astronautics; 1996:469-483. doi:10.2514/5.9781600866388.0469.0483.
10. Zeng H, Park H, Fontaine MD, Smith BL, McGhee KK. Identifying Deficient Pavement Sections by Means of an Improved Acceleration-Based Metric. *Transp Res Rec J Transp Res Board*. 2015;2523:133-142. doi:10.3141/2523-15.
11. Seraj F, Zhang K, Turkes O, Meratnia N, Havinga PJM. A smartphone based method to enhance road pavement anomaly detection by analyzing the driver behavior. In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers - UbiComp '15*. New York, New York, USA: ACM Press; 2015:1169-1177. doi:10.1145/2800835.2800981.
12. Mukherjee A, Majhi S. Characterisation of road bumps using smartphones. *Eur Transp Res Rev*. 2016;8(2):13. doi:10.1007/s12544-016-0200-1.
13. Douangphachanh V, Oneyama H. Formulation of a simple model to estimate road surface roughness condition from Android smartphone sensors. *IEEE ISSNIP 2014 - 2014 IEEE 9th Int Conf Intell Sensors, Sens Networks Inf Process Conf Proc*. 2014;(April):21-24. doi:10.1109/ISSNIP.2014.6827694.
14. Chen S-Y, Shih A, Hsiao C-Y. Road conditions detection using Arduino based sensing module and smartphone. In: *2015 IEEE International Conference on Consumer Electronics - Taiwan*. IEEE; 2015:254-255. doi:10.1109/ICCE-TW.2015.7216884.
15. Bello-Salau H, Aibinu A, Onwuka E, et al. A New Measure for Analysing Accelerometer Data towards Developing Efficient Road Defect Profiling Systems. *J Sci Res Reports*. 2015;7(2):108-116. doi:10.9734/JSRR/2015/16840.
16. Rajamohan D, Gannu B, Rajan K. MAARGHA: A Prototype System for Road Condition and Surface Type Estimation by Fusing Multi-Sensor Data. *ISPRS Int J Geo-Information*. 2015;4(3):1225-1245. doi:10.3390/ijgi4031225.
17. Joubert JW, de Beer D, de Koker N. Combining accelerometer data and contextual variables to evaluate the risk of driver behaviour. *Transp Res Part F Traffic Psychol Behav*. 2016;41:80-96. doi:10.1016/j.trf.2016.06.006.
18. Hanson T, Cameron C, Hildebrand E. Evaluation of low-cost consumer-level mobile phone technology for measuring international roughness index (IRI) values. *Can J Civ Eng*. 2014;41(9):819-827. doi:10.1139/cjce-2014-0183.
19. Chugh G, Bansal D, Sofat S. Road Condition Detection Using Smartphone Sensors : A Survey. *Int J Electron Electr Eng*. 2014;7(6):595-601. http://www.ripublication.com/irph/ijeeev_spl/ijeeev7n6_09.pdf.
20. Wirth R, Hipp J. CRISP-DM: Towards a Standard Process Model for Data Mining. In: *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*. ; 2000:29-39. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.198.5133&rep=rep1&type=pdf>. Accessed May 15, 2017.
21. Ghose A, Biswas P, Bhaumik C, Sharma M, Pal A, Jha A. Road condition monitoring and alert application: Using in-vehicle Smartphone as Internet-connected sensor. In: *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*. IEEE; 2012:489-491. doi:10.1109/PerComW.2012.6197543.