

PADS: A Reliable Pothole Detection System Using Machine Learning

Jinting Ren and Duo Liu^(✉)

College of Computer Science, Chongqing University, Chongqing, China
jintingren@gmail.com

Abstract. In recent years, with the popularity of private cars, an increasing number of people prefer autos as their way to travel. However, poor road conditions may cause damages to vehicles and have drawn great concern of governments all over the world. The extreme weather conditions, heavy traffics and low road quality are worsen the situation, making it is a challenging task to keep roads in good conditions. Therefore, frequent repairs are required to avoid damages to vehicles. In this paper, we propose a reliable pothole detection system using machine learning (PADS) to facilitate the road pothole detection and road conditions maintenance. The proposed system provides low latency in potholes detecting, thereby shortening the time for road maintainers to identify poor conditions roads. To make our system easy to deploy, we reduce monetary costs and simplify system architecture. To improve accuracy in potholes detection, we use *K-MEANS* algorithm based on basic threshold algorithm. Our results display a plot of z-axis accelerations on one road and a pothole-marked map. At last, we show the pothole detection accuracy comparison between our algorithm and basic threshold algorithm.

Keywords: Pothole detection · Road monitoring · Machine learning · IoT · Tri-axial accelerometer

1 Introduction

Poor road surface conditions may cause damages to vehicles, high maintenance cost and even traffic accidents. According to the survey of U.S. Federal Highway Administration [1], thousands of people are hurt or killed each year on roads and highways due to poor road quality and conditions. Therefore, keeping roads in good conditions is important to reduce the traffic accidents. However, maintaining road surface is a challenging task due to extreme weather condition, heavy traffic and the low road quality. To maintain good road condition, frequent repairs are required and make that a reliable and low-latency road conditions monitoring system is much required. This kind of system will be especially useful in reminding road maintainers to repair poor road surfaces.

However, traditional monitoring systems are not suitable in such scenario for the following two reasons: (1) Traditional monitoring systems aim at a overall monitor on roads conditions including pavement images collection, pavement shape detection, road stereo image drawing and etc. Often, this kind of overall

monitoring will cost lots of time and can not be low-latency apparently [2,3]. (2) Traditional monitoring systems collect data from two main sources—statistical sensors in the pavement and cars equipped with specific road monitoring sensors like Ground Penetrating Radar (GPR) [4]. Limited by cost, these systems can not scale up to monitor most roads in the city.

In this paper, we proposed PADS a simplified, low-cost and reliable road monitoring system with low latency in detecting potholes. Considering the drawbacks of traditional systems and needs of keeping roads in good conditions, there are three main concerns in our system – achieving low latency in detecting potholes, reducing monetary cost and improving detection accuracy. First, we make only pothole detection function left in our system to achieve low-latency goal. The number of Potholes on road is a convincing evidence to identify whether a road is in good conditions and we can use this feature to get a general picture of road conditions. We can also use the potholes location report to help road maintainers fix poor road surfaces quickly. Second, PADS is implemented using ‘dynamic’ sensors — sensors placed on autos to reduce cost [5]. Due to the mobility, autos can be randomly distributed in city roads and we can monitor almost every road theoretically if we have enough autos equipped with sensors. Using ‘dynamic’ sensors instead of statistical sensors on roads can reduce monetary cost because placing sensors on several vehicles is cheaper than placing sensors on every road. Finally, wrong potholes detection will make system unauthentic and increase the maintenance cost. To improve detection accuracy, our approach decides to use machine learning methods [6,7]. The *K_MEANS* algorithm [8,9] is used to cluster road data into two classes and then computes a more accurate threshold to improve detection accuracy. For evaluation, we show a pothole-marked map produced by our system and compare the accuracy of our potholes detecting algorithm with basic threshold algorithm.

In summary, we make the following major contributions:

- We propose a reliable pothole detection system with low latency in detecting potholes.
- Our design uses machine learning methods to improve the accuracy in detecting potholes compared of threshold-based pothole detection algorithm.
- We deploy our system on private cars and detect potholes on roads in low latency.

The rest of this paper is organized as follows: Sect. 2 describes the background and the motivation. Section 3 discusses on our system design including hardware platform and architecture. Main algorithm description and its complexity analysis are also included in this section. Section 4 talks about the evaluation of our approach consisting of a potholes marked map, a plot of z-axis accelerations on one road and accuracy analysis. In Sect. 5, the related work will be discussed on. At last, we draw a conclusion in Sect. 6.

2 Background and Motivation

In this section, we first introduce the dangers caused by poor road conditions. We then give the drawbacks of traditional monitoring systems and explain why it can not be applied to keep roads in good conditions. Finally, we discuss our motivation.

2.1 Dangers Caused by Poor Road Conditions

The U.S. Congress passed in a rare bipartisan effort (late 2015) the Surface Transportation Reauthorization and Reform Act of 2015, which provides \$233 billion for federal highway maintenance over five years. That is \$46 billion per year [10]. The fact shows that every year governments need to spend much money on maintaining road conditions. However, the road conditions of U.S is still poor even though it has cots government so much to maintain road conditions. According to the research, in the cities with worst road conditions in U.S, the ration of poor roads is over 50%. It is a really challenging task to keep roads in good conditions but governments need to do this because poor road conditions can cause many dangers.

There are several dangers that may caused by poor road conditions. First of all, hitting a pothole may cause damages to vehicles. The damages may not only occur in automotive chassis but also in tire puncture and wire rim. In most cases, these damages may just make you pay for repair charge. However, the worse case is traffic accidents. According to statics, one-third accidents involve poor road conditions of approximately 33,000 traffic fatalities each year [10]. Moreover, the number of accidents is predicted to become larger with increases in vehicle traffic in next years.

2.2 Drawbacks of Traditional Monitoring Systems

Traditional monitoring systems is not useful in helping road maintainers fix road surfaces rapidly. These systems are designed to do an overall check on roads. Therefore, it will include many unnecessary check for just fixing the road surface and result in long latency. Moreover, traditional monitoring system often use sensors placed on roads or specific monitoring cars. Limited by the cost, tradition monitoring systems are hard to scale up and then can not monitor most roads in cities. For example, one current equipment used in measuring road condition, which is composed by accelerometers, distance measuring instruments and graphic displays is quite expensive [11]. This road condition systems may cost 8,000 to 220,000 dollars. These two drawbacks determine that traditional road monitoring system can not be applied in fixing road surfaces rapidly to keep roads in good conditions.

2.3 Motivation

Private vehicles have been more and more popular in recent years. To guarantee driving safety, governments take frequent repairs to keep roads in good conditions. Therefore, how to identify roads in poor conditions quickly is important since road maintainers cannot repair any poor conditions roads without identifying them. Fortunately, a road monitoring system may help road maintainers achieve this goal. However, traditional road monitoring systems are long-latency and high-cost. Therefore, they can not be applied to identify poor condition roads quickly. This motivates us to propose a reliable pothole detecting

with low latency in detecting potholes. To reduce the cost, we choose only 3-axis accelerometers sensors and GPS module as our data-collecting equipment. To improve detection accuracy, we use machine learning methods to compute a more accurate threshold in identifying potholes.

3 PDSML Design

In this section, we first discuss the hardware platform used by our system. We then introduce the architecture of our system. Finally, we talk about the main algorithm and its complexity analysis.

3.1 Hardware Platform

We reduce monetary costs especially those spent on hardware to make our system easier to scale up. To achieve this goal, only cheap sensors and inexpensive boards are used in our system. One acceleration sensor and one GPS module are used in collecting raw data. For preprocessing raw data and sending filtered data to center servers, we use raspberry pi as our router. We choose raspberry rather than other boards due to its relatively higher computing performance as we decide to run filtering algorithm on router. Another benefit of preferring raspberry is that raspberry board has a built-in wifi module for sending data and has no need for any external data transmitting module. The last component of our system hardware platform is a common center data server. Ignoring the cost of the server, the total hardware platform cost is less than \$60, a acceptable price comparing with the cost of traditional road condition monitoring systems. Figure 1 shows the top view of our hardware platform.

3.2 Software Architecture

PADS uses a basic IoT architecture consisted of four layers [12]. Figure 2 shows architecture of our system. The first layer is the sensing layer contains acceleration sensors which are used to collect accelerations in 3-axis and locations respectively. Routers (i.e., raspberry pi 3) in second layer — the network layer — receive raw data and filter out noisy data. After that, routers will send filtered

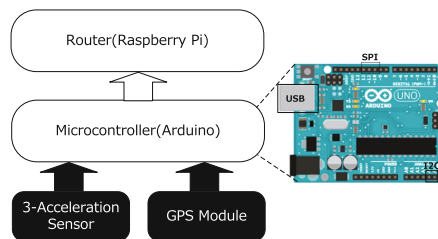


Fig. 1. Top view of our hardware platform.

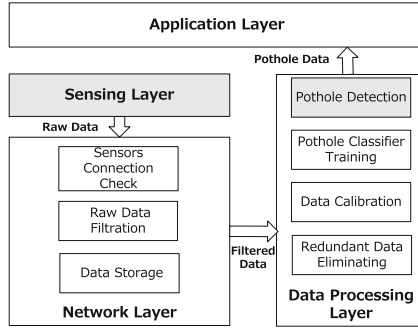


Fig. 2. Architecture of PADS.

data to center data server using built-in wifi module. The most critical part of our system is the third layer — the data processing layer. In this layer, PADS will remove replicated data since we have deployed multiply device on roads. With deduplicated dataset, we still have to ‘calibrate’ it. Because our equipment is not placed flat on front tyres, our location of potholes will be inaccurate and we need to ‘calibrate’ the location data. Data-collecting latency and the speed of car equipped our sensors will also affect the accuracy in pothole location. Taking all these factors into account, we ‘calibrate’ our data to eliminate errors. Once our system get enough simplified and accurate data, a potholes classifier can be trained and then we can detect potholes using this classifier. At last, in application layer we mark all the detected potholes on Google map it the APIS provided by Google.

3.3 Data Processing

There are four main algorithms in our system. However, the filtering algorithm, a Z-DIFF algorithm (using the difference value in Z-axis acceleration to judge if this data should be ignored), is so simple that we have no need to discuss on it in detail. Thus, algorithm part will be mainly divided into three parts—redundancy solving algorithm part, calibrating algorithm part and pothole detection algorithm part.

Redundant road data eliminating algorithm: The first part is about redundant road data eliminating algorithm. This algorithm is used in reducing redundant data in same position collected by different vehicles or one vehicle in different periods. We use distance calculated by longitude and latitude data to judge if a new data is redundant. The computational formula is:

$$\begin{aligned}
 a &= \sin^2(\Delta\varphi/2) + \cos(\varphi_1) \cos(\varphi_2) \sin^2(\Delta\lambda) \\
 c &= 2 * \operatorname{atan2}(\sqrt{a}, \sqrt{1-a}) \\
 \text{Distance} &= R * c.
 \end{aligned} \tag{1}$$

In above equation, φ denotes latitude, λ denotes longitude and R denotes the earth's radius. However, computing distance every time a new data comes can be a time-wasting task especially when searching space becomes large. To achieve low latency in detecting potholes, shrinking searching room is necessary. We give the road label to each data. When a new 'normal' data (no visual z-axis acceleration changes) comes to server, PADS get its road information using map APIS and check if the system has already stored the data of this road. Our system will ignore the data if the road condition information has been stored in our system. Only the 'normal' data can be ignored in our system and every data that possibly represents pothole will be stored in database. Unfortunately, though system will spend less time compared with computing distance for every data, our system still needs to search road information when a new data comes.

Since searching road information every time is still a time-consuming task, our system has to find a new way to label our data. In fact, vehicles may often stop in the crossroad and the speed of the vehicle will become zero. As a result, a zero-speed data collected by GPS may represent that the vehicle has arrived at the crossroad. The idea is simple, PADS will store the 'normal' data in buffer at first without searching its road information (we will search the road information for the first data) until a new data that has zero value in speed comes. Then PADS compares the road information between the first data and the 'zero-speed' data. After that, our system will ignore the 'normal' data between them if they are on the same road. The drawback of this solution is that we still have to search the road information of each data when the zero-speed data is actually collected in another road. However, the worst case will cost nearly the same time compared with unoptimized algorithm.

Calibrating algorithm: The second part is about calibrating algorithm. Calibrating algorithm is used in making location more accurate. For calibrating data, we should take data-collecting latency and speed into consideration to fix the measurement error caused by the distance between seats and front tyres. We can computer the direction of vehicles using three-axis accelerations. Then we can use the direction and the distance to get the accurate location. However, this direct method can also be time-consuming. We need a simpler method to achieve the low-latency goal. Based on the fact that the distance is so short and we can assume the speed and the direction of the vehicle keep steady during this period. With that assumption, we can replace the location information of current data with one of later collected data to get a more accurate location. All our system do is to compute the latency for our vehicles to drive the distance between seats and front tyres. We define ν for the speed of the vehicle and ψ for the frequency of GPS in collecting data (GPS module have the same frequency as acceleration sensors in collecting data). The τ represents the distance between seats and front tyres. The latency can be expressed as:

$$Latency = \left\lfloor \frac{\psi\tau}{\nu} \right\rfloor. \quad (2)$$

With latency computed, we can calibrate our data.

Pothole detection algorithm: The last part is about pothole detection algorithm. The simplest way to detect pothole on road is using a threshold algorithm [13]. However, assuming that roads in different areas have their own features, using one certain threshold to detect potholes in all areas is not realistic. We solve this problem by building a more accurate model for potholes detecting using machine learning [14]. Indeed, using machine learning to detect pothole will affect performance and make latency longer. We still need to prove accuracy because inaccurate pothole detection will make system unauthentic. In that context, a simple but suitable machine learning algorithm should be chosen to balance accuracy and latency. Finally, our system chooses the K-MEANS algorithm [15]. Using this algorithm, we can simply cluster the data into two clusters using three features x-axis, z-axis and speed. In related work, the threshold only uses z-axis as its feature. However, a pothole will cause changes both in x-axis acceleration (decrease in x-axis acceleration) and z-axis (increase in z-axis acceleration) Using two features will make detection more accurate. We do not use speed directly to cluster data. In contrast, we define two scenes for clustering—high speed scene and low-speed scene. Since acceleration data changes will be quite different between high-speed and low-speed scenes, we need to treat them separately (in our experiment, we define over 25 kmph is high-speed and below it is low-speed). Supposed potholes data as abnormal data, it should be smaller in number of two clusters. However, the results show the two clusters often have similar size. To remedy this issue, we use original threshold (only use z-axis data as threshold) [13] to firstly label the data (pothole label and normal label) and then cluster them. After clustering, we choose the data set having more pothole labels to be the cluster represents pothole data. Then we use the x-axis and z-axis accelerations data of pothole data set to calculate our detection threshold. The computational formula is:

$$Threshold = \frac{\sum_{i=1}^n (\chi - z)^2}{n}. \quad (3)$$

Let χ denotes x-axis acceleration and z denotes z-axis acceleration. We use z to minus χ in equation because large z-axis acceleration and small x-axis acceleration mean a pothole.

3.4 Complexity Analysis

The most time-consuming part of our algorithm is pothole detection part. To reduce latency, our system decides to update pothole threshold periodically. In most cases, PADS just uses computed threshold to judge if the vehicle has faced a pothole. Therefore, in that situation, the complexity of detecting algorithm is $\mathcal{O}(n)$. When the system needs to update threshold, the complexity of detecting potholes is mostly dependent on the complexity of *K_MEANS_CLUSTER* algorithm [16, 17]. The general Euclidean space d and clusters k will decide *K_MEANS_CLUSTER*'s complexity. For our system, both d and k equal to 2 (i.e. 2 attributes *x_axis* and *z_axis*). Therefore, the complexity is $\mathcal{O}(n^5 \log n)$. Though the performance of *K_MEANS_CLUSTER* is

Algorithm 3.1 Pothole detection algorithm

Input: *train_data_vector*:road condition data vector;*SIZE*:size of road condition vector;*OT*:original threshold provided by other work;*ST*:speed threshold

Output: *Threshold_vector*:pothole threshold in high-speed and low-speed

```

1: cluster  $\leftarrow$  2, iterations  $\leftarrow$  100. //one cluster for pothole data and another
   for normal data
2: cluster1_vector  $\leftarrow$   $\emptyset$ , cluster2_vector  $\leftarrow$   $\emptyset$ , TH  $\leftarrow$  0.
3: for i  $\leftarrow$  1 to SIZE do
4:   if train_data_vector[i].z_axis > OT then
5:     train_data_vector[i].label  $\leftarrow$  pothole.
6:   else
7:     train_data_vector[i].label  $\leftarrow$  normal.
8:   end if
9: end for
10: result  $\leftarrow$  K_MEANS_CLUSTER(train_data_vector, cluster, iterations).
11: if result[0].pothole_label_count > result[1].pothole_label_count then
12:   cluster1_vector  $\leftarrow$  result[0].
13: else
14:   cluster1_vector  $\leftarrow$  result[1].
15: end if
16: high_speed_sum  $\leftarrow$  0.
17: low_speed_sum  $\leftarrow$  0.
18: n  $\leftarrow$  length(cluster1_vector).
19: for i  $\leftarrow$  1 to n do
20:   if cluster1_vector[i].speed > ST then
21:     high_speed_sum  $\leftarrow$  high_speed_sum + (cluster1_vector[i].z_axis -
       cluster1_vector[i].x_axis)2.
22:   else
23:     low_speed_sum  $\leftarrow$  low_speed_sum + (cluster1_vector[i].z_axis -
       cluster1_vector[i].x_axis)2.
24:   end if
25: end for
26: Threshold_vector.add( high_speed_sum/n).
27: Threshold_vector.add( low_speed_sum/n).
28: return Threshold_vector

```

not so good, the complexity of detecting algorithm is acceptable since updating thresholds occurs in low frequency.

4 Evaluation

In this section, we firstly describe the experimental environments to evaluate our system. Secondly, we display a pothole-marked map and a plot of z-axis accelerations on one road as our experimental results. At last, we compare the accuracy between our detection algorithm and basic threshold algorithm.

Table 1. Item model and detailed features of components

Name	Model	Interface	Resolution	Sample rate	Accuracy	Power supply
3-axis accelerometer	ADXL345	SPI I2C	up to 16 g	up to 3200 HZ	4 mg/LSB	2.0 V to 3.6 V
GPS module	UBLOX NEO-6M	UART USB SPI DDC(I2C compliant)	N/A	up to 5 Hz V	2.5 m CEP	2.7 V to 3.6 V
Logic level converter	5 V to 3.3 V Logic Level Converter	N/A	N/A	N/A	N/A	5 V or 3.3 V
Data collecting board	Genuino UNO	UART USB SPI I2C TWI	N/A	N/A	N/A	5 v
Router	Raspberry Pi 3	UART USB SPI I2C WIFI	N/A	N/A	N/A	2.5 A@5 V

4.1 Experimental Setup

As shown in Table 1, we use ADXL345 and UBLOX NEO-6M to comprise our sensing layer. We use Arduino as a middle data collecting board instead of connecting sensors directly to raspberry due to the limit of raspberry pins amount. Another significance of this design is making system loosely-coupled. Separating router and sensing layer can make it easier to replace sensors without modifying codes on router. We use SPI protocol to connect board with ADXL345 and I2C protocol to connect board with GPS module. The benefit of using two different protocols is to avoid conflicts on pins using. Without sharing pins, we can easily implement our system using only one board. The ADXL345, however, can not be directly connected to Arduino due to its 5 V-system. Therefore, a logical level converter is necessary to convert 5 V to 3.3 V. At last, we connect Arduino to raspberry with a USB cable. Fixing all our equipment on the seat of vehicle, we will finish the setup for our experimental platform.

4.2 Results

The user interface of our system is a pothole-marked map on web page. Moreover, a plot of vehicle's accelerations data on one road which uses sensor data directly has been drawn as a by-product. Figure 3 shows the z-axis accelerations during the vehicle drive on one road. In the figure, time represents data collecting time. From the figure we can see there visual changes on vehicle's z-axis accelerations.

To evaluate our system, we first use z-axis accelerations data and its road information to find which area may have more potholes. We optimize the workload to improve experimental efficiency [18]. After that, we drive the car equipped with experimental equipment on this area for several times. Using these

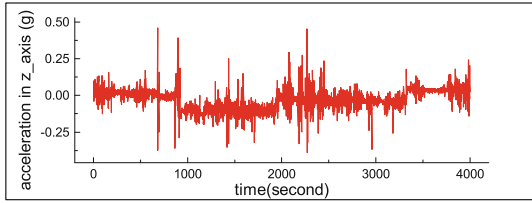


Fig. 3. A plot of *z-axis* accelerations on one road.

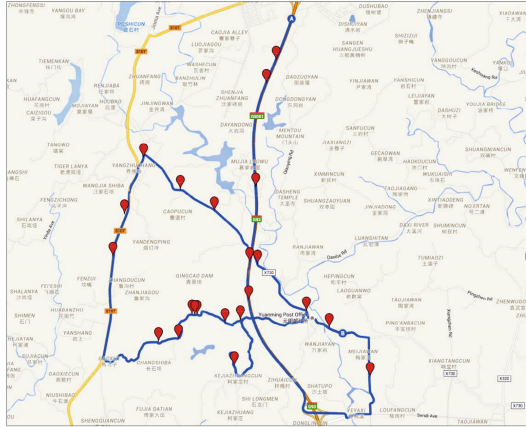


Fig. 4. A pothole-marked map which represents the road surface conditions near G93 highway.

data, our system detects potholes in this area. To make our map more clear, we eliminates all potholes that do not belong to main stem by hand. Finally, we get a pothole-marked map. Figure 4 shows the final pothole-marked map produced by our system. In that map, we have marked potholes detected by our system around G93 highway with read markers.

After analysing results, we find that our system actually can not distinguish pothole from occasional road anomalies on roads like emergency brake. In fact, our system can only distinguish potholes from normal road data. However, these occasional road anomalies have little chance to occur in the same place for several times. Therefore, our system only take the potholes that has been detected for several times to distinguish potholes from occasional road anomalies.

At last, we will evaluate detecting accuracy of our system. The simplest but most time-consuming way to get test data set is using a field trip. However, we are prohibited to stop and check potholes on many roads. To solve it, we drive our test car on certain area roads for several times. Then, we take these potholes data that have been detected every time as actual pothole on road to consist our test data set. Let $N_{non_pothole_detection}$ be the number of the pothole that has not been detected and $N_{pothole_wrong_detection}$ be the number of normal road data that has been judged pothole wrongly. $N_{detection}$ represents the total

number of test data set. We define accuracy of detecting potholes as:

$$Accuracy = 1 - \frac{N_{non_pothole_detection} + N_{detection}}{N_{detection}} \quad (4)$$

The results show that our pothole detection algorithm has truly improved detecting accuracy compared with simple threshold algorithm in z-axis.

5 Related Work

Road Quality Clustering System. Automatic road quality clustering system has ranked road quality utilizing tri-axial accelerometer [19]. This system has used accelerometer data to cluster roads into three classes dependent on its quality. However, it can not locate the potholes on road and has little use in fixing road surface quickly.

Pothole Detection Systems Based on Smart Phones. Current pothole detection systems using vehicles equipped with sensors are often based on smart phones. These systems have used tri-axial accelerometer and GPS of smart phones as our equipment on board to detect potholes [13, 20–23]. However, this kind of systems need more human interaction and can not be automatic. Moreover, differences in mobile phones hardware make systems based on it hard to develop.

Pothole Detection using Threshold Algorithm. The simplest way to detect potholes on roads is using threshold algorithm in z-axis [13]. Detecting potholes dependent on certain threshold is simple and has lowest latency. However, the accuracy of this kind of detection is doubted. Moreover, the certain threshold can not be applied to different road conditions.

6 Conclusion

This paper has described the design, implementation and evaluation of a pothole detection system. We have used 3-axis accelerometer sensors and GPS module to collect raw data. Using inexpensive sensors and boards, our equipment costs only about \$60. To filter data, we have used a simple Z-DIFF algorithm. We have already solved the redundancy of collected data and calibrated measurement errors. To improve detection accuracy, we have used machine learning methods. We have analysed the complexity our detection algorithm. In evaluation part, we have displayed a pothole-marked map produced by our system and shown z-axis accelerations of one road. Finally, we have evaluated the accuracy of detecting potholes by comparing it with simple threshold algorithm. Overall, we have developed a reliable pothole detection with low latency in detecting potholes.

Acknowledgements. This work is partially supported by grants from the National Natural Science Foundation of China (61672116, 61601067), Research Fund for the Doctoral Program of Higher Education of China (20130191120030), Chongqing High-Tech Research Program cstc2016jcyjA0332, Fundamental Research Funds for the Central Universities (CDJZR14185501, 0214005207005), Chongqing University (2012T0006).

References

1. Federal Highway Administration (1966). <https://www.fhwa.dot.gov/>
2. Cucchiara, R., Piccardi, M., Mello, P.: Image analysis and rule-based reasoning for a traffic monitoring system. *IEEE Trans. Intell. Transp. Syst.* **1**(2), 119–130 (2000)
3. Takagi, M., Masaki, I.: Road surface condition monitoring system using sensors disposed under the road, 27 August 2002. US Patent 6,441,748
4. Roadscanners. Roadscanners oy (1998). <http://www.roadscanners.fi/>
5. Gai, K., Qiu, M., Zhao, H., Tao, L., Zong, Z.: Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. *J. Netw. Comput. Appl.* **59**, 46–54 (2016)
6. Rudin, C., Wagstaff, K.L.: Machine learning for science and society. *Mach. Learn.* **95**(1), 1–9 (2014)
7. Kodratoff, Y., Michalski, R.S.: *Machine Learning: An Artificial Intelligence Approach*, vol. 3. Morgan Kaufmann, San Francisco (2014)
8. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S., et al.: Constrained k -means clustering with background knowledge. In: *ICML*, vol. 1, pp. 577–584 (2001)
9. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k -means clustering algorithm. *J. Roy. Stat. Soc. Ser. C (Appl. Stat.)* **28**(1), 100–108 (1979)
10. POTHOLE (2002). <http://www.pothole.info/>
11. Budras, J.: A synopsis on the current equipment used for measuring pavement smoothness (2001). <http://www.fhwa.dot.gov/pavement/smoothness/rough.cfm>
12. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of things (IoT): a vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **29**(7), 1645–1660 (2013)
13. Prashanth Mohan, R.R., Padmanabhan, V.N.: Nericell: rich monitoring of road and traffic conditions using Mobile Smartphones (2008)
14. Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco (2005)
15. Jain, A.K.: Data clustering: 50 years beyond k -means. *Pattern Recogn. Lett.* **31**(8), 651–666 (2010)
16. Velmurugan, T., Santhanam, T.: Computational complexity between k -means and k -medoids clustering algorithms for normal and uniform distributions of data points. *J. Comput. Sci.* **6**(3), 363 (2010)
17. Ghosh, S., Dubey, S.K.: Comparative analysis of k -means and fuzzy c -means algorithms. *Int. J. Adv. Comput. Sci. Appl.* **4**(4), 34–39 (2013)
18. Gai, K., Du, Z., Qiu, M., Zhao, H.: Efficiency-aware workload optimizations of heterogeneous cloud computing for capacity planning in financial industry (2015)
19. Hsu, J. Y.-J., Tai, Y.-C., Chan, C.-W.: Automatic road anomaly detection using smart mobile device (2010)
20. Zviedris, R., Kanonirs, G., Selavo, L., Mednis, A., Strazdins, G.: Real time pothole detection using android smart-phones with accelerometers (2011)
21. Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., Balakrishnan, H.: The pothole patrol: using a mobile sensor network for road surface monitoring. In: *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pp. 29–39. ACM (2008)
22. 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 (PERCOM Workshops)*, pp. 489–491. IEEE (2012)
23. Friedlander, R.R., Kraemer, J.R.: Evaluating road conditions using a mobile vehicle, 17 September 2013. US Patent 8,538,667