

A Fused Method of Machine Learning and Dynamic Time Warping for Road Anomalies Detection

Zengwei Zheng^{ID}, Mingxuan Zhou^{ID}, Yuanyi Chen^{ID}, Meimei Huo^{ID}, Lin Sun^{ID}, Member, IEEE,
Sha Zhao^{ID}, Member, IEEE, and Dan Chen^{ID}

Abstract—To discover the condition of roads, a large number of detection algorithms have been proposed, most of which apply machine learning methods by time and frequency processing in acceleration and velocity data. However, few of them pay attention to the similarity of the data itself when the vehicle passes over the road anomalies. In this article, we propose a method to detect road anomalies by comparing the data windows with various length using Dynamic Time Warping(DTW) method. We propose a model to prove that the maximum acceleration of a vehicle passing through a road anomaly is linear with the height of the road barrier, and it's verified by an experiment. This finding suggests that it is reasonable to divide the window by threshold detection. We also apply a brief random forest filter to roughly distinguish normal windows from anomaly windows using the aforementioned theory, in order to reduce the time consumption. From our study, a system is proposed that utilizes a series of acceleration data to discover where might be anomalies on the road, named as Quick Filter Based Dynamic Time Warping (QFB-DTW). We show that our method performs clearly beyond some existing methods. To support this conclusion, experiments are conducted based on three data sets and the results are statistically analyzed. We expect to lay the first step to some new thoughts to the field of road anomalies detection in subsequent work.

Index Terms—Threshold detection, sliding window, accelerometer, dynamic time warping, machine learning.

I. INTRODUCTION

ROAD anomalies can lead to serious traffic accidents. For example, between 2000 and 2011, there were 2 million traffic accidents in Canada, of which 33% were related to road conditions or bad weather. In 2015, about 50,000 British drivers reported road crashes causing traffic accidents, and road pits caused a car accident every 11 minutes. As a result, governments spend huge amounts of manpower and resources on road maintenance. The British government announced that

Manuscript received November 21, 2018; revised June 20, 2019, December 23, 2019, and May 21, 2020; accepted July 30, 2020. Date of publication August 26, 2020; date of current version February 2, 2022. This work was supported in part by the Young Scientists Fund of the National Natural Science Foundation of China under Grant 61802343; in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LGF19F020019 and Grant LGN20F020003; and in part by the Intelligent Plant Factory of Zhejiang Province Engineering Laboratory. The Associate Editor for this article was C. Guo. (*Corresponding author: Yuanyi Chen*)

Zengwei Zheng, Mingxuan Zhou, Yuanyi Chen, Meimei Huo, Lin Sun, and Dan Chen are with the School of Computer and Computing Science, Zhejiang University City College, Hangzhou 310015, China (e-mail: chenyuanyi@zucc.edu.cn).

Sha Zhao is with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China.

Digital Object Identifier 10.1109/TITS.2020.3016288

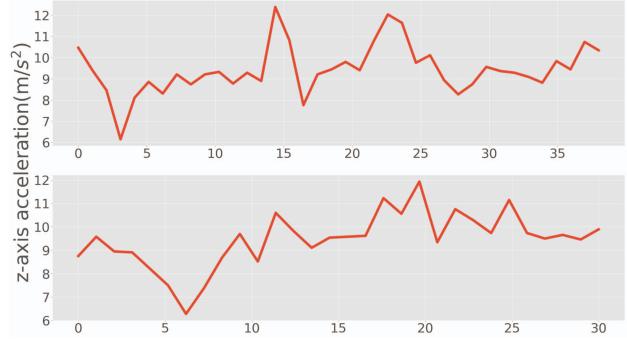


Fig. 1. Comparison of z-axis data when passing over two metal bumps.

it spent \$1.2 billion on road maintenance in 2017. In 2014, for the city of Toronto, Canada spent a total of \$6 million on road repairs. Therefore, detecting road anomalies in an efficient and simple way is very important.

Traditional road detection techniques have many limitations and consume a lot of manpower and resources [1]. With the development of mobile devices and sensor technologies, it is possible to detect roadway surface disruptions (anomalies [2]) using data collected by sensors carried by mobile devices. An anomaly is any permanent obstacle generated by the continuous use, weather conditions, or traffic planning decisions in the road. As the vehicle travels through the anomaly, the acceleration of the vehicle in different directions changes, and this change is collected by the acceleration sensor of the devices. So the problem is converted into discovering possible anomalies from sensor data.

Early work [3]–[5] mainly found the anomalies hidden in the sensor data by means of threshold detection. For example, [3]–[5] detected bumps and potholes, respectively. Although these methods can detect whether there is an anomaly, they cannot distinguish the type of anomalies. Furthermore, the detection accuracy is relatively low.

In the past few years, many studies have extracted different kinds of features [2], [5]–[9], and used machine learning methods [2], [7], [9], [10] such as Support Vector Machine, k-means or decision tree. The drawbacks of these works are: 1) they rarely rule out the impact of normal roads on model training; 2) As the Figure 1 shows, the data between two anomalies in the same one category may be similar in time domain, but none of them take this into consideration and 3) they often use a fixed window length, leading to the possibility to slice anomalies' data.

Taking these issues into consideration, this article focuses on: (1) realizing a more realistic and meaningful situation where once a series of acceleration data is given (to simulate a series of data is collected in the reality), a set of dynamic length anomalies and their types are determined, (2) proposing a new data set and (3) to obtain a good performance in the data set compared to some methods proposed in other papers.

Our model is able to detect and distinguish three kinds of anomalies: pothole, speed bump and metal bump. We utilize some existing data set and also collect data to validate that the effectiveness of our method in different kinds of data set or different situations. Thus our work can be the basis for subsequent road inspections such as formulating a road anomaly map of the city.

The contributions of this work are three-folds:

- We propose a novel model to analyze the relationship of the maximum of z-axis data and the height or depth of an anomaly, and conduct an experiment to prove this theory briefly.
- A method of threshold detection to define the window and a coarse classifier to separate normal types from exception types are proposed based on the theory. The former one locate the probable position that might exist anomalies, and the latter one reduces time consumption and improve performance.
- A method with both high performance and acceptable time consumption is introduced to classify specific types of anomalies. Compared with some existing methods, our method performs much better in F1 score while it takes only twice the time consumption of the method with the least time consumption.

The organization of the article is as follows. Section II reviews some related works during the past few years. Section III gives the main ideas and implementation of our method. Section IV presents the experimental results for multi-class anomalies classification, whose main indicators are F1 and time. Finally, Section V concludes our work and predicts possible research directions.

II. RELATED WORK

The problem of classifying roadway surface disruption based on data collected from mobile devices is tackled roughly in early research, mainly using threshold detection techniques to identify the anomalies. Smartphones *et al.* [3] proposed a system using acceleration information of the smart-phone at a sampling rate of 310 Hz. Eriksson *et al.* [4] presented a system that collected more than 9730 kilometers of acceleration and GPS data at a sampling rate of 380 Hz and 1 Hz in the Boston area, detecting road damage using a threshold-based filter. The system proposed by Mednis *et al.* [5] used four heuristic threshold methods to detect road damage: Z-THRESH, G-ZERO, Z-DIFF and STDEV. The threshold-based detection method has the following problems: it can only detect whether the road is damaged, but cannot identify the type of roadway surface disruption, and the detection accuracy is not high.

In recent years, many researchers believe that the combination of time domain and frequency domain information extraction features can more accurately help to solve the problem, and some researchers processed data based on the assumption that it can be modeled by machine learning methods. The approaches are divided into two paradigms.

From the perspective of feature extraction, Perttunnen *et al.* [6] used the fast Fourier transform and the Mel frequency cepstral coefficient to obtain the energy value of each frequency band of the acceleration information, and extracted a 95-dimensional feature for detection. Seraj *et al.* [7] utilized wavelet transform to decompose acceleration data to extract feature vectors. The research result of Carlos *et al.* [8] showed that the identification feature based on the standard deviation score can obtain better detection effect. El-Wakeel *et al.* [9] used wavelet denoising method to improve the quality of low-cost MEMS sensor sensing data, and made a roadway surface disruption based on the characteristics of time domain and frequency domain information extraction of acceleration. Gonzlez *et al.* [2] took a word bag model to extract features from the acceleration data. In [11], they used some features such as the final stop duration or the number of stops of the car to characterize the road condition. In [12], [13], they took the angle of the road into consideration to modify the acceleration data. Li *et al.* [14] proposed a new metric to value the acceleration data. From the perspective of methods, Support Vector Machine [9], [13], [15], [16], k-Means [2], [15], decision trees [17], [18], k-nearest neighbours [19] were mostly used. Also, some existing work built model with equations to process data. In [20], they proposed some equations to use the acceleration data to discover the dimension of the pothole. Alessandroni *et al.* [21] researched the relationship between the road roughness and the vehicle speed using their own model. In [22], they used 288 features per window to calculate the Euclidean distance. In [23], they established a rough judgement between the height or depth of road anomalies and the type of road anomalies.

One of the drawbacks of these studies is that firstly they do not think highly of the quantity difference between the normal road and the anomaly. According to data collected by some studies [17], [24], the ratio between normal road and anomaly is about 95:5 to 97:3. As a result, sliding window techniques used in [24] generate a large amount of windows that represent normal road, and it makes the procedure of SVM slow and inaccurate. Some methods [17] used imbedded platform to filter those normal road data, but it's complicated in design and cannot be applied to all mobile devices. In [25], they used Gaussian model to predict the depth of the potholes. In [26], they employed under-sampled oscillation system to estimate the road surface.

Another drawback is that the existing work extracts some features to reflect and represent the most important characteristic of the window, but none of the existing work discovers the fact that the data collected when a vehicle has driven over the anomaly is similar not only in features but also in the shape itself.

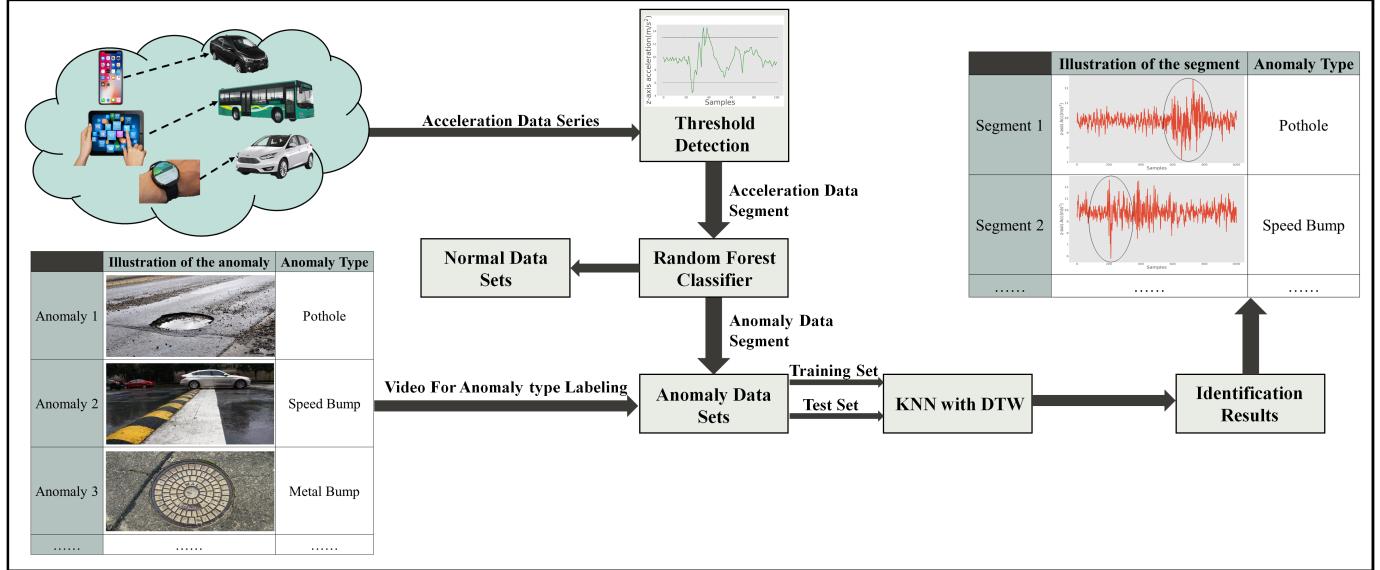


Fig. 2. The framework of the proposed anomaly detection and identification model.

TABLE I
RELATED WORKS AND THEIR MAIN CHARACTERISTICS

Ref.	What was detected	Technique	Performance
Nericell [3]	Bumps	Threshold	8% FP, 41% FN
Mednis [5]	Potholes	Threshold	92% TP
Pothole Patrol [4]	Potholes	Threshold	92.4% accuracy
Perttunen [15]	Anomalies	SVM	3% FP, 18%FN
Word bag [8]	Anomalies	word bag expression	88% accuracy
Astarita. [17]	Anomalies	others	85.6% accuracy
Seraj et al. [7]	Potholes	Threshold	93% TP
Fox et al. [13]	Bumps	Threshold	85.6% TP

The last disadvantage is that from a practical point of view, using a fixed window size in sliding window technique [24] or make each anomaly included in a fixed time window after some processing [17] may encounter the following issues: the division method can slice the anomaly into two windows, then the information of the anomaly decreases in both window; or the length of window can be less than the length of the anomaly data, then the anomaly cannot be perfectly covered by one window.

Table I provides a summary of the related works and their characteristics.

III. OVERVIEW OF THE PROPOSED MODEL

A. Model Framework

Our proposed model produces the detection and judgement result of an existing piece of data by two phases: 1) Threshold detection and sliding window processing; 2) A quick filter of the normal window and the abnormal window and 3) Comparison to determine types using DTW and KNN. The framework is shown in Figure 2.

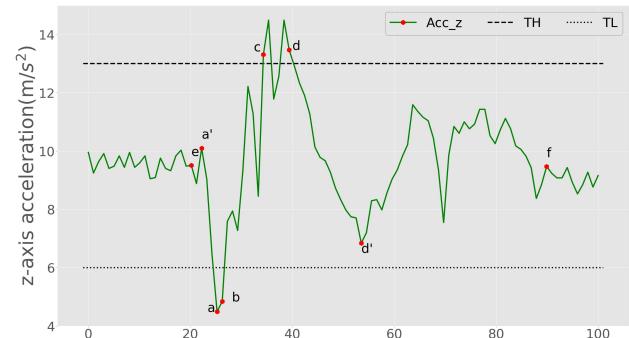


Fig. 3. Illustration of threshold detection and sliding window.

Our method directly locates where there might exist some anomalies using threshold detection and sliding window technique as candidate windows. From these candidate windows, we apply a fast random tree filter with features extracted from the acceleration data image and the anomalies. This step is meant to roughly separate those normal windows located as anomalies from those anomalies windows in actual. Finally, we apply the Dynamic Time Warping technique to classify the remaining windows into determined types.

B. Threshold Detection and Sliding Window Processing

Figure 3 shows an example of the procedure of threshold detection and sliding window algorithm, and the Algorithm 1 shows the algorithm of a normal judgement. This procedure is based on two facts that are observed throughout the data set:

- If the vehicle drives in a normal road, the data of z-axis acceleration floats in a certain range with a Gaussian noise.
- Once driving over an anomaly, the data of z-axis acceleration often firstly exceeds a high threshold and then falls under a low threshold; or it behaves in an opposite

Algorithm 1 Threshold Detection and Sliding Window

Require: A series of acceleration data $D = (x_1, y_1, z_1), (x_2, y_2, z_2), \dots, (x_m, y_m, z_m)$. Given the threshold TL, TH ; the size of window; the probable detection range.

- 1: Set $k = 1$
- 2: **while** $k \leq m$ **do**
- 3: **if** $z_k \geq TH$ or $z_k \leq TL$ **then**
- 4: Detection Range = $[k, k + Range]$
- 5: Set $k' = k - Window, k'' = k' - Window, z_{avg1} = avg([z_{k'}, \dots, z_k]), z_{avg2} = avg([z_{k''}, \dots, z'_k])$
- 6: **if** $z_k \geq TH$ **then**
- 7: **for** $i = k + 1$ to $k + Range$ **do**
- 8: Right boundary $r = \max i$ that satisfy $z_i \leq TL$
- 9: **end for**
- 10: **while** $z_{avg1} > z_{avg2}$ **do**
- 11: Update the average by (2)
- 12: **end while**
- 13: **else**
- 14: **for** $i = k + 1$ to $k + Range$ **do**
- 15: Right boundary $r = \max i$ that satisfy $z_i \geq TH$
- 16: **end for**
- 17: **while** $z_{avg1} < z_{avg2}$ **do**
- 18: Update the average by (2)
- 19: **end while**
- 20: **end if**
- 21: Set $r' = r + Window, r'' = r' + Window, z_{avg1} = avg([z_{r'}, \dots, z_r]), z_{avg2} = avg([z_{r''}, \dots, z'_r])$
- 22: **if** $z_k \geq TH$ **then**
- 23: **while** $z_{avg1} < z_{avg2}$ **do**
- 24: Update the average by (4)
- 25: **end while**
- 26: **else**
- 27: **while** $z_{avg1} > z_{avg2}$ **do**
- 28: Update the average by (4)
- 29: **end while**
- 30: **end if**
- 31: Add final window = $[k', r'']$ to candidate window set C
- 32: $k = r''$
- 33: **else**
- 34: $k = k + 1$
- 35: **end if**
- 36: **end while**
- 37: **return** Result Set C

way, that is firstly falling under a low threshold, then passing a high threshold. After this large vibration, it may continue to fluctuate in value, and eventually fall back to the normal position.

With these two preliminaries, we suppose once the threshold point is detected, we can find the point at which the z-axis acceleration data starts to change, and the point at which the z-axis acceleration data ends to normal. Our candidate window will be located between these two points, which is probably an anomaly.

We take a pothole and its data as an example. It's supposed to start at point e and end at point f , shown in Figure 3.

Firstly, point a is found for the reason that the value of z-axis acceleration exceeds the high threshold TH and its x-axis coordinate x_a is marked. We set a parameter $Range$ for the reason that once a point over the high threshold is found, there should be at least one point that falls below the low threshold, according to our preliminaries. We suppose the last point that satisfies will cover the entire anomaly best, so we find it in a big enough range to cover most of anomalies in order not to miss any anomaly. Then two points a and d are found.

These two points are found by threshold detection. But there are two points in fact that don't cover the entire anomaly, and we'll lose information if we take them as candidate windows. We set the left window range begin as $[x'_a, x_a]$, where $x'_a = x_a - Window$, and the average z-axis acceleration of this range is marked as z_{avg} . Then the window will continuously updated by Equation 1

$$x'_a = x'_a - Window, x_a = x_a - Window \quad (1)$$

and the average will be updated by:

$$\begin{aligned} k' &= k' - Window, k'' = k'' - Window, \\ z_{avg1} &= z_{avg2}, z_{avg2} = avg([z_{k''}, \dots, z'_k]) \end{aligned} \quad (2)$$

until one z_{avg} is smaller than the former one. Then the left end of the candidate window will be marked as the final x'_a . The reason is the z-axis data will often rise or decline from the normal value to pass over high threshold or below low threshold. If the average value of the sliding window changes in a single direction (always rise or decline), we suppose the sliding window doesn't reach the end yet. As a result, the sliding window might be a bit bigger than the actual anomaly, but we suppose it's acceptable if the window size is not too big. We perform the similar operation to the right end point d only changing the upgrade equation by Equation 3,

$$x'_d = x'_d + Window, x_d = x_d + Window \quad (3)$$

and the average will be updated by:

$$\begin{aligned} r' &= r' + Window, r'' = r'' + Window, \\ z_{avg1} &= z_{avg2}, z_{avg2} = avg([z_{r''}, \dots, z'_r]) \end{aligned} \quad (4)$$

until one z_{avg} is bigger than the former one, and obtain the right end of the candidate window as the final x'_d .

Without loss of generality, for anomalies that z-axis data firstly decline and then incline, the operation will be similar and the only thing that need to be changed is to adjust the situation in which the cycle ends as the result of the change of z_{avg} .

C. A Quick Filter of the Normal Window and the Abnormal Window

Windows in the candidate window set are not all windows generated from anomalies data. Only 1/4 of the total windows are abnormal windows. In order to exclude this part of the normal window, we observe some data and study the work of others, and formulate a random forest model. The features are shown in Table II.

The vehicle can be regarded as a single degree of freedom vibration model during driving. The equation to describe the

TABLE II
VARIOUS FEATURES AND THEIR DEFINITIONS

Feature	Definition
average	Average value of the z-axis acceleration data of the window
max	The number of maxima points in a time window
min	The number of minima points in a time window
var	Variance value of the z-axis acceleration data of the window
max mean diff	Difference between the max and the average
mean min diff	Difference between the average and the min

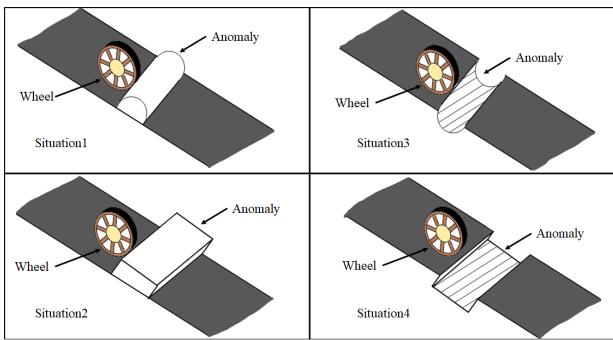


Fig. 4. Illustration of four simple situations. Situation 1 and situation 2 are bumps, while situation 3 and 4 are potholes.

vibration in the vertical direction of the single degree of freedom system is:

$$m\ddot{z} = -k[z(t) - r(vt)] \quad (5)$$

where m is the quality of the vehicle, k represents the elasticity coefficient of the system, t is the driving time of the vehicle, v is the speed of the vehicle, and $r(x)$ describes obstacles on irregular roads mathematically. Assume that the vehicle is initially moving at a constant speed, that is, $z(0) = 0$, $\dot{z}(0) = 0$. The maximum reaction force value on the road after the vehicle passes through the obstacle is $kz(t)$, which means when $t > T$ (suppose the vehicle passes through the vehicle after T), $r(vt)$ will be constantly 0 and the system is free to vibrate. Then the maximum reaction force is $F = kZ$, and Z is the amplitude of free vibration:

$$Z = \sqrt{z^2(T) + [\dot{z}(T)/\omega_0]^2} \quad (6)$$

where $\omega_0 = \sqrt{k/m}$ to be the fundamental frequency of the system.

To illustrate (5), we formulate four cases to simulate those anomalies such as potholes, metal bumps and speed bumps. They are shown in Figure 4 and are simplified in mathematics. Take the situation 1 as an example. Use the equation

$r(x) = \frac{a}{2}(1 - \cos\frac{2\pi x}{b})$ [27], (5) can be solved as:

$$z(t) = \frac{a\omega_0^2}{2(\omega_0^2 - \omega_p^2)} \cos\omega_p t - \frac{a}{2} + \left[\frac{a}{2} - \frac{a\omega_0^2}{2(\omega_0^2 - \omega_p^2)} \right] \cos\omega_0 t \quad (7)$$

where $\omega_p = 2\pi v/b$ to be the vibration frequency of the system. Also, $z'(t)$ can be calculated:

$$z'(t) = -\frac{a\omega_0^2}{2(\omega_0^2 - \omega_p^2)} \omega_p \sin\omega_p t - \left[\frac{a}{2} - \frac{a\omega_0^2}{2(\omega_0^2 - \omega_p^2)} \right] \omega_0 \sin\omega_0 t \quad (8)$$

When $t = T$ and according to $\omega_p T = 2\pi$, we can get:

$$z(T) = \frac{a\omega_0^2}{2(\omega_p^2 - \omega_0^2)} [1 - \cos\omega_0 T] \quad (9)$$

and:

$$z'(T)/T = \frac{a\omega_p^2}{2(\omega_p^2 - \omega_0^2)} \sin\omega_0 T \quad (10)$$

Bring (9) and (10) into (6), we can get:

$$Z = \frac{a\omega_p^2}{(\omega_p^2 - \omega_0^2)} \sin \frac{\omega_0 T}{2} \quad (11)$$

Situation 2 will reach the same result. As for situation 3 and 4, a similar result will be:

$$Z = a \sin \frac{\omega_0 T}{2} \quad (12)$$

Taking the speed into consideration, we can get:

$$Z = dv/dt + a \sin \frac{\omega_0 T}{2} \quad (13)$$

where v is the traveling speed and t is the contact time between the vehicle and the anomaly. According to [27], the dv/dt changes from 10 to 16, depending on different types of suspension systems of the car or the shape of anomaly. Our data proves this threshold can be finely set to detect most of anomalies.

We utilized our electrical motor to do this experiment. We made some board with 0.2cm height to simulate the experience of driving over the bump. We placed board on board to reach the height we set. Then we fixed one mobile phone which runs the acceleration data collecting application on the electrical motor to avoid the shift of phone and drove over the board. We drove for 5 times on each height to acquire the mean value. For the depth relation, we found several potholes in the road with different depths and did the same measurement. As a result, we reach the conclusion that the variance of z-axis acceleration when the vehicle drives through the anomaly will have a linear relationship with the depth or height of the anomalies. We conduct some experiments to illustrate this finding. The results are shown in Figure 6 and 5, representing the situation1 and situation 4, respectively. It can be seen that the depth or height of the anomalies has a positive correlation with the magnitude of the acceleration, and to some extent is a linear relationship. Based on this conclusion,

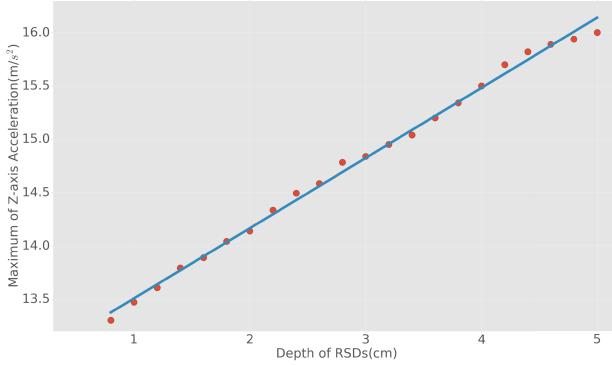


Fig. 5. Relationship between the depth of anomaly and maximum acceleration.

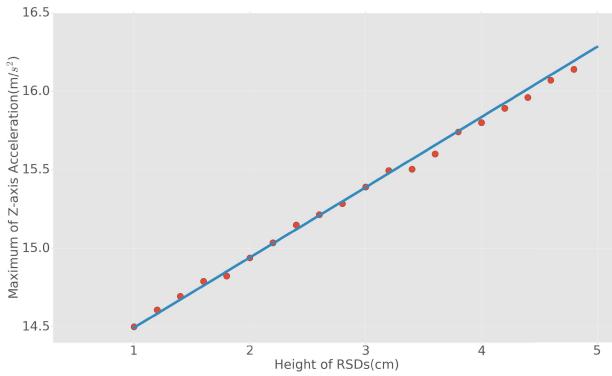


Fig. 6. Relationship between the height of anomaly and maximum acceleration.

we mainly select some indicators that can reflect the change of acceleration as features to identify anomalies from normal road. Another reason is that the fixed window, and not all abnormal windows contain max or min in their methods. Often an anomaly window contains some points that belong to normal road, making the average closer to normal, and thus the variance becomes smaller for anomaly window. As a result, the difference between extremum and average decreases. Our method does not have the effect of fixed window methods on features. The window would cover almost all information, thus the features become significant.

D. Comparison to Determine Types Using DTW and kNN

This step is mainly based on the idea of kNN algorithm and DTW algorithm. From the former abnormal candidate window set AC , select one candidate window and mark it as WC_i . Then the data vector of this window is $VC_i = \{(x_1, x_2, \dots, x_n), (y_1, y_2, \dots, y_n), (z_1, z_2, \dots, z_n)\}$ where n represents the length of candidate window and x, y, z represents the x,y,z-axis acceleration data respectively. We choose one window WT_j from the training window list and the data vector is $VT_j = \{(x_1, x_2, \dots, x_m), (y_1, y_2, \dots, y_m), (z_1, z_2, \dots, z_m)\}$ where m represents the length of training window and its label is L_j .

We apply the DTW algorithm to these two window respectively on three axis acceleration data, and get three distances D_x, D_y, D_z . We simply figure out the mean distance

by calculate the average of $D_{ij} = (D_x + D_y + D_z) * 1/3$. Repeat this step and we will get a series of distances between candidate window WC_i and the training window set $D_i = \{D_{i1}, D_{i2}, \dots, D_{il}\}$ where l is the numbers of training windows. We get a series of labels $L = \{L_1, L_2, \dots, L_j\}$ as well. We sort the distance vector D_i to get k smallest distance. We can figure out the numbers of anomalies from label vector L , and suppose we find n_1, n_2, n_3 potholes, speed bumps and metal bumps that have the smallest distance to WC_i , which means they are similar to WC_i . We figure out the biggest number in n_1, n_2, n_3 and mark the label of WC_i as the label of it.

IV. EXPERIMENT EVALUATION

In this section, we will present the results of a series of experiments conducted to evaluate the performance of the proposed anomaly detection model. Firstly, we describe the settings of experiments including data sets, compared methods and evaluation metrics. Then, we will report and discuss the experiment results.

A. Experimental Settings

1) *Data Sets:* Data Set 1: We develop an application running in Android to collect mobile device sensor data including GPS, acceleration and velocity. Besides, video is taken as well to identify the position of anomalies and add labels. One device is tied to the rear of an electric motor to collect data. We run on about 10 roads and finally achieve a 15-minute data set size. The speed of the vehicle is kept at about 30km/h. We only run 1 time over each road for the thought of road anomaly non-repeatability. In order to maintain the uniformity of anomaly types, we marked three types of anomalies according to the data set attached to [24]: potholes, metal bumps and speed bumps.

Data set 2: This data set is based on the anomalies data that the paper [24] provides. In order to expand the scale of the problem to meet more realistic scenarios, we use all the anomalies data and normal road data, randomly generate a road that contains every anomaly included in the data set. To simulate the real road in the real world, we insert anomalies into normal road in a realistic proportion according to the ratio of anomalies to normal road in existing data sets [2], [28], [29]. The ratio of the length of the anomalies to the normal road is set to 1:50.

Data set 3: The third one is inspired by the paper [30] to simulate and collect data from Carsim®. We use the Carsim program to simulate vehicles driving over potholes, metal bumps and speed bumps. With this tool, we simulated vehicles driving over 2000 potholes, 1200 metal bumps and 1200 speed bumps. A large amount of acceleration data is collected.

The size of three data sets is shown in Table III.

2) *Comparative Methods:* We compare the proposed recommendation model with the following four methods, all of which are popular work from the last three years. One of them is for single-class anomaly recognition, that is, to classify pothole from normal road. The rest of them is to not only

TABLE III
DATA SETS AND THE SIZE OF ANOMALIES

Data Set	Amount of Pot-hole	Amount of Speed bump	Amount of Metal Bump
Data Set 1	82	17	32
Data Set 2	236	138	135
Data Set 3	2000	1200	1200

classify the anomalies but also perform anomalies and normal road identification:

- **Support Vector Machine For Who Is Who (SVM-WW):** SVM-FWW [24] was originally proposed to identify different anomalies such as potholes, metal bumps and speed bumps based on z-axis data acceleration. Features are mainly some statistical indicators such as mean and standard deviation. They also proposed confidence score as feature to value whether the statistical features are trustworthy.
- **Cloud Pothole Detector With Self Feature Selection(CPD-SFS):** CPD-SFS [13] was originally proposed to detect different lanes and to identify anomalies in different lanes. They managed to select most effective features by greedy forward selection algorithm, and create some features that are more complicate such as the average of the absolute value of the product of z-axis acceleration and the velocity.
- **Support Vector Machine based on Multi-Domain Data Processing(SVM-MDDP):** SVM-MDDP [9] was originally proposed to identify various anomalies via sensor data based on multi-domain of processing. They first de-noised the data, then not only extracted statistical features like other literature, but also used time and frequency domain data, and eventually achieved a feature scale of more than 70 dimensions.
- **Machine Learning Method based on Word Bag (MLM-WB):** MLM-WB [2] was originally proposed to represent the time series data with some simplified representations called word bag. Basing on these representation, machine learning methods reach a higher accuracy.

We compare these methods with our method, **Quick Filter Based Dynamic Time Warping(QFB-DTW)**. The results will be shown in the follow section.

3) *Evaluation Metric:* There are two main indicators for our evaluation: F1-score and runtime. For F1-score, it's calculated by the equation:

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (14)$$

In order to explain the problem more clearly, and to make our model closer to practical applications, we identify a TP as the overlapping between the prediction and the real data [24], as the Figure 7 shows. The ground truth indicates the anomaly in the real world, and the predicted line 1 and 2 are generated by some methods. The predicted line 1 and the ground truth overlap in time series 10 to 15, so we suppose the prediction is

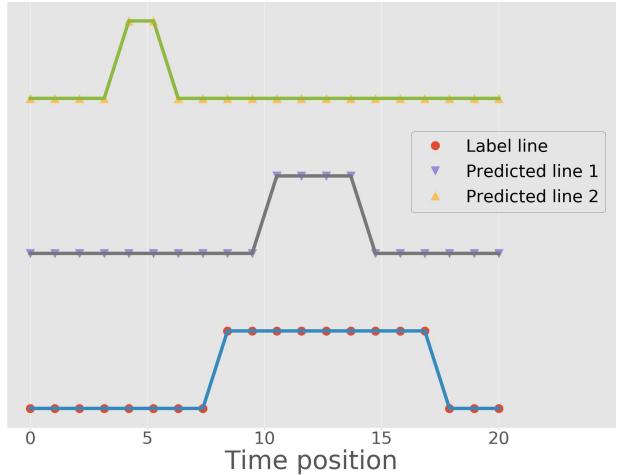


Fig. 7. Illustration of overlapping between prediction and ground truth to be a True Positive while predicted line 1 is marked as anomaly and predicted line 2 is marked as normal way.

correct and it's a TP. The predicted line 2 and the ground truth don't overlap in any position, but there is a prediction, so it's a FP. In the real prediction situation, we want to know if there is an anomaly on the road. We also want to know the specific location of the anomaly. However, predicting the true length of the anomaly is often unnecessary: the road maintenance department does not care about this type of information. As a result, this definition of TP is more reasonable. From an operational point of view, other methods use the idea of sliding windows, so we think of their feature window as an object that compares whether it overlaps. Our method is to cut the window and compare it with the original data. FP, FN have similar definitions.

From the perspective of runtime, the DTW method requires a lot of time while achieves higher accuracy. In order to reduce the running time consumption, we adopt a fast random forest to greatly reduce the number of windows to be compared, and some optimization of the DTW algorithm is implemented. Suppose two anomaly windows are similar in shape. One may have the length of 50 and the size of the other one may be 200. Because they are similar in shape, it's impossible to get the minimized distance using DTW algorithm if we compare the 10 20 dots in the former window with the 160 180 dots in the latter window. As a result, if two anomaly windows are similar, to minimize the distance between them, the searching route by DTW method will be among the diagonal as Figure 8 shows.(If they are not similar, this step won't influence the result because we don't care how small the result is.)

In the end, although there are not significant differences in the running time between our method and other methods, the index of F1-score of our method is much higher.

B. Experimental Results

1) *Impact of Model Parameters:* Once the data set is given, our method will automatically tune the parameters according to the data set. Here we will use data set 2 to be an example of tuning the parameters.

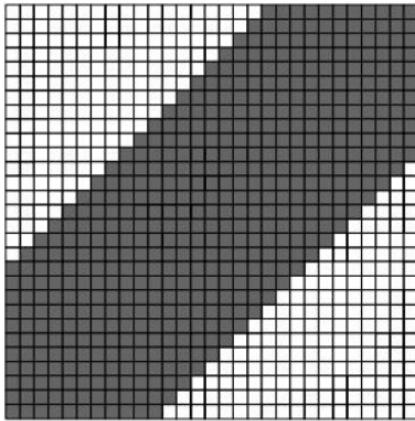


Fig. 8. Illusion of the restriction in searching region.

TABLE IV
IMPACT OF THRESHOLD TO F1 SCORE

TL \ TH	10.8	10.9	11.0	11.1	11.2	11.3	11.4	
TL	7.8	0.878	0.859	0.867	0.875	0.885	0.909	0.904
7.9	0.876	0.831	0.856	0.878	0.859	0.881	0.903	
8.0	0.890	0.843	0.877	0.878	0.897	0.893	0.892	
8.1	0.860	0.848	0.863	0.919	0.899	0.914	0.928	
8.2	0.853	0.864	0.896	0.878	0.915	0.904	0.912	
8.3	0.866	0.869	0.936	0.904	0.910	0.901	0.926	
8.4	0.874	0.868	0.897	0.906	0.897	0.922	0.894	
8.5	0.877	0.855	0.873	0.876	0.893	0.906	0.926	
8.6	0.890	0.882	0.890	0.911	0.901	0.911	0.885	
8.7	0.876	0.853	0.872	0.874	0.889	0.864	0.889	
8.8	0.881	0.828	0.849	0.864	0.873	0.887	0.872	
8.9	0.863	0.875	0.845	0.845	0.877	0.848	0.853	

Firstly, we adjust the threshold to well fit the data set. The way in which the data is collected, equipment, vehicles, etc., result in differences in the magnitude of the acceleration data. Thus it's necessary to find a good threshold to help the model detect anomalies precisely. It's reasonable that we cannot set the low threshold too low or the high threshold too high, or the model will ignore a great amount of possible anomalies. As a result, the model adjust the low threshold TL from 7.1 to 9.5, and the high threshold TH from 10.8 to 12, which are suitable ranges that F1 score drops sharply because of misdetection. The model will use 70% of the data set to train and 30% of the data set to test and reach a brief threshold. As an example, we use the data set 2 to illustrate the importance of threshold adjustment and the results are shown in Table IV. The F1 score is the average of pothole F1, metal bump F1 and speed bump F1 to show the result more directly. The table shows the influence of a high threshold or a low threshold. The model achieves the best performance when high threshold is set to 11 and low threshold is set to 8.3.

After setting the threshold, we still adjust the k in kNN step and window size in threshold detection step. A large window size leads to a slow convergence speed, while a small window will lead to information loss and cannot cover the entire anomaly. The influence of k in kNN lies in whether the data in the training set is sufficiently representative that

TABLE V
IMPACT OF WINDOW SIZE W AND K TO F1 SCORE

k \ w	1	2	3	4	5	6	7
1	0.882	0.889	0.883	0.885	0.881	0.869	0.883
2	0.835	0.852	0.838	0.823	0.826	0.843	0.864
3	0.810	0.846	0.826	0.827	0.826	0.815	0.829
4	0.792	0.815	0.810	0.813	0.822	0.814	0.799
5	0.737	0.774	0.785	0.801	0.788	0.781	0.768
6	0.766	0.759	0.798	0.781	0.785	0.758	0.778
7	0.735	0.774	0.770	0.782	0.746	0.721	0.760
8	0.745	0.753	0.772	0.772	0.755	0.739	0.763
9	0.723	0.744	0.784	0.758	0.743	0.749	0.739
10	0.740	0.764	0.751	0.759	0.722	0.721	0.750

as many anomalies as possible are correctly identified. The results are shown in Table V. The table shows that the data in the data set is representative enough and the window size should be set to 2. These parameters, such as thresholds and window size, will keep same once the data set remains same in the 10-fold cross validation step.

2) *Identification Performance for All Methods:* After the tuning step, we locate some reasonable parameters. Using these parameters, the model set a 10-fold cross validation and reach the final detecting results.

The results of data set 1, data set 2 and data set 3 are illustrated in Table VI. Data set 1 is collect by motor bike, and there only exist a small amount of speed bumps and metal bumps, compared to the other two data sets. As a result, there may not exist enough metal bumps or speed bumps in the validation part, leading to a bad performance for all the methods.

From data, we can observe: 1) All the other methods perform better in pothole identification reaching more than 0.5 in F1 score. The best method SVM-MDDP achieves a F1 score more than 0.6. Compared to these methods, our method always passes over 0.8 and achieves at least 0.2 advantage in F1 score. 2) In speed bump identification, Compared to other methods, our method reaches 12% better than SVM-MDDP. 3) In metal bump identification, our method still performs better than other methods. Method SVM-MDDP reaches 0.660. Compared to this method, our method performs 22% better.

For this data set, there are two features that require special instructions. Firstly, the data is all collected by motor bikes, which means the fluctuation of the acceleration data is much larger than the data that's collected by cars when the vehicle passes over the same anomaly. Similarly, there will be more noise in the acceleration data for the same reason. Secondly, there are less speed bumps and metal bumps compared to potholes. As a result, F1 score is greatly affected by the TP in detection. Often a TP will have a large fluctuation to F1 score. This situation is especially obvious with high training set partition.

Despite this influence, we find method SVM-MDDP performs especially well in pothole detection and metal bump detection. It's because: 1) this method utilizes more than 70 dimensions of features, making it covers more information. 2) the feature of this method is more complex compared to other methods for its features covering statistical, time and

TABLE VI
F1 SCORE OF DATA SETS WITH 10-FOLD CROSS VALIDATION

Data sets	Anomaly Type	Methods	QFB-DTW	SVM-WW	CPD-SFS	SVM-MDDP	MLM-WB
Data set 1	Pothole	0.900	0.591	0.548	0.682	0.580	
	Speed Bump	0.667	0.561	0.576	0.622	0.599	
	Metal Bump	0.800	0.543	0.540	0.660	0.592	
Data set 2	Pothole	0.921	0.679	0.628	0.665	0.614	
	Speed Bump	0.909	0.609	0.653	0.741	0.619	
	Metal Bump	0.880	0.693	0.631	0.752	0.674	
Data set 3	Pothole	0.939	0.677	0.844	0.829	0.717	
	Speed Bump	0.874	0.654	0.787	0.789	0.728	
	Metal Bump	0.819	0.731	0.800	0.779	0.701	

frequency domain. They reflect more characteristics while it also takes a large amount of time to data pre-processing. It'll be shown in the next part.

We can also find all the other methods perform bad in metal bump or speed bump prediction. The reason of this is firstly, all these methods require a fixed length of window. However, due to the difference in vehicle speed, an appropriate window for data sets that're collected by car cannot be fit for this data set. Secondly, the lack of speed bumps and metal bumps in the training set has a great influence on the model training. Although it's not quite universe to collect data by motor bikes, we can still conclude that the imbalance between different anomalies, even the imbalance between normal road and the anomalies may lead to low recognition rate. Our method still performs better in such imbalanced situation.

Data set 2 includes a fine proportion of pothole, speed bump and metal bump. Besides, the number of anomalies are much bigger than the data set 1, which leads to a more steady result.

From the data, we can observe: 1) All the other methods perform better in pothole identification reach more than 0.6 F1 score. The best method SVM-WW reaches 0.679. Compared to these methods, our method passes over 0.9 and achieves at least 0.2 advantage in F1 score. 2) In speed bump identification, method SVM-FWW and MLM-WB perform around 0.6, while method SVM-MDDP achieves higher than 0.7. Compared to the best method SVM-MDDP, our method has a 18% most advantage. 3) In metal bump identification, our method still performs better than other methods. Method SVM-MDDP reaches beyond 0.7. Compared to this method, our method achieves 16% better.

This data set is used by method SVM-WW, and its performance improves quite obviously compared to the result of data set 1. Its F1 score of speed bump almost reaches 0.6, and 0.4 for metal bump identification. The paper indicates that it does not quite fit for multi-class identification, as the reason that the statistical feature of this method such as standard, mean or the covariance may be similar for all the anomalies. As a result, this method cannot distinguish different anomalies from each other. Method CPD-SFS takes some complex features into consideration, such as the product of the velocity and the z-axis acceleration data. The complexity of the features contributes to the identity of metal bumps and potholes with speed bumps. However, for the similarity between the metal bumps and the speed bumps, such as the

vehicle both rising first and the z-axis data beginning similarly. This method requires sufficient samples to train the model and improve the performance. Method SVM-MDDP performs better than the other methods for the reason mentioned before. Our method absorbs the advantage of other methods. That is, to fully utilize the advantage of machine learning methods to briefly divide the normal road from the anomalies. This step is quite reliable according to the previous analysis in Section III A and the result of other methods. After that, the reliable result will be produced by DTW algorithm avoiding the default of existing machine learning methods that they can hardly distinguish different anomalies.

Data set 3 is collected by simulation, which means the data is more smooth and regular. Thus, performance of all the methods improve compared to data set 1 and data set 2.

From the data, we can observe: 1) All the other methods perform better in pothole identification reach more than 0.6 F1 score. The best method CPD-SFS and SVM-MDDP reach more than 0.8. Compared to these methods, our method passes over 0.9 and achieves almost 0.1 advantage in F1 score, and achieves 11% better than the best of other methods. 2) In speed bump identification, method SVM-MDDP still performs best in the other methods. Compared to this method, our method achieves 13% better. 3) In metal bump identification, the result of our method and the best performance of other methods is very close. Our method only takes a 0.02 advantage in F1 score.

This data set is generated by a simulation software, causing its noise interference to be much smaller than the other two data sets, as a result the data set data is more regular. In this situation, some methods that have made ways to eliminate noise, such as SVM-WW with some features as the confidence of the other features, have lost the role of these features to some extent. The others, such as the methods of CPD-SFS and SVM-MDDP, dedicate to the processing of the data in numerical, time or frequency domain, take the possession of high F1 score. On the other hand, the performance of these two methods in other data sets containing noise is not satisfactory as well. From this perspective it might not be suitable to take the noise adjustment into feature consideration. Our method avoids this problem with the concept that once the data set is fairly big enough, there will exist a fairly similar window with similar influence of the noise, making it unaffected from the noise.

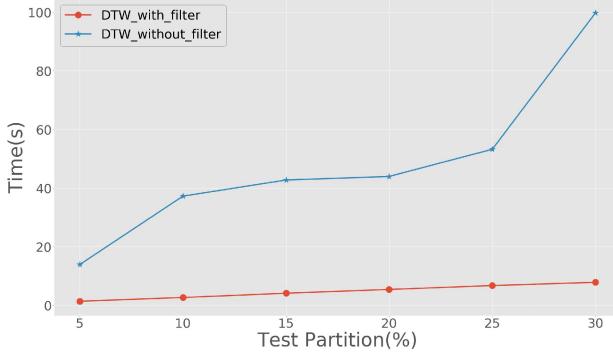


Fig. 9. Influence of the random forest step to time consumption.

3) Time Consumptions for All Methods:

a) *Hardware environment*: The program runs in Windows platform PC with a Intel i7-7700k cpu and 16GB ddr4 ram. We refer to the paper [24] and use python package ski-learn to formulate our program. All the basic methods such as random forest or support vector machine are all programmed with that package. We didn't use GPU to accelerate the calculation or the model training. Python programs run in python 3.5.

b) *Influence of random forest step*: We take the data set 2 to figure out the influence of the random forest step. We use 70% of the data set to train and increases the size of the validation set from 5% to 30% by 5%. Each partition we randomly choose the validation part and run the program by 5 times. The result is shown in Figure 9. It can be seen that the time consumption is around 10 times without the random forest filter compared to the time with it. We investigate the composition of the window and find only 1/4 of them are anomaly windows, while the remain of them are all false detection. In Dynamic Time Warping algorithm, each window in the validation set will be calculated with each window in the training set, making it a $O(n^2)$ relationship to the total size of the windows n . Theoretically, the time consumption will be reduced to about 1/16. Some windows are mixed to the other side, leading to a certain degree of decline in performance. However, since our method performs quite well in anomaly detection, we suppose it is acceptable to change the algorithm at the expense of performance in exchange for a significant reduction in time.

c) *Time consumption analysis*: The results of time consumption are illustrated in Figure 10, Figure 11, Figure 12. We still use 70% of the data set to train and increase the size of validation set from 5% to 30% by 5%. To reach a more accurate result, we do prediction by 5 times with a fixed model. The time consumption includes the pre-processing time, which means transforming data into feature windows, and prediction time, which means the time consumption for prediction. From the three figures, we can observe: 1) Time consumption grows almost linearly with the size of validation part. Also, the relative order between these methods remain unchanged. Thus, it's understandable that the order of time consumption between these methods stay unchanged despite the size of the data set. 2) Method MLM-WB and SVM-FWW spend the least time, while the method SVM-MDDP spends

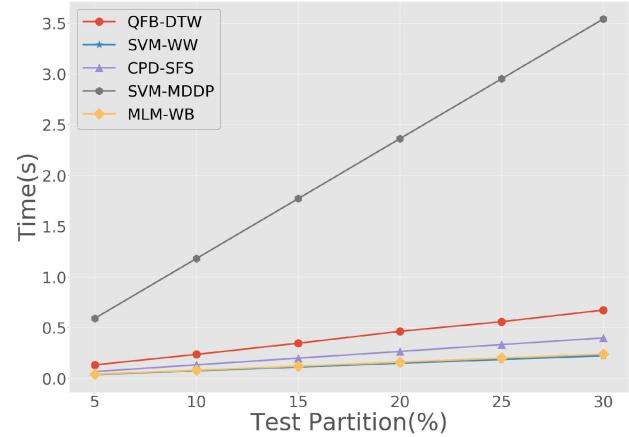


Fig. 10. Time consumption of data set 1.

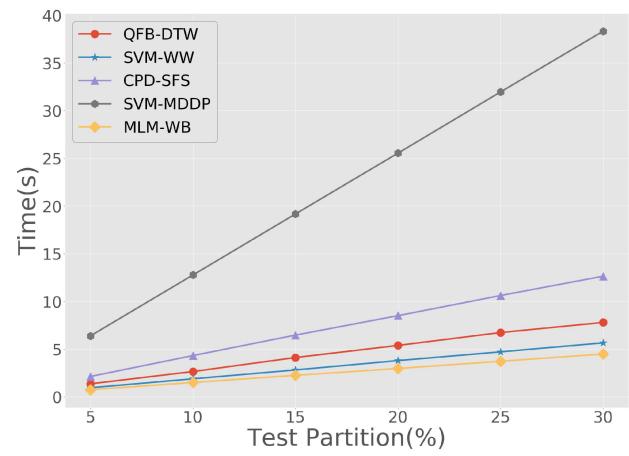


Fig. 11. Time consumption of data set 2.

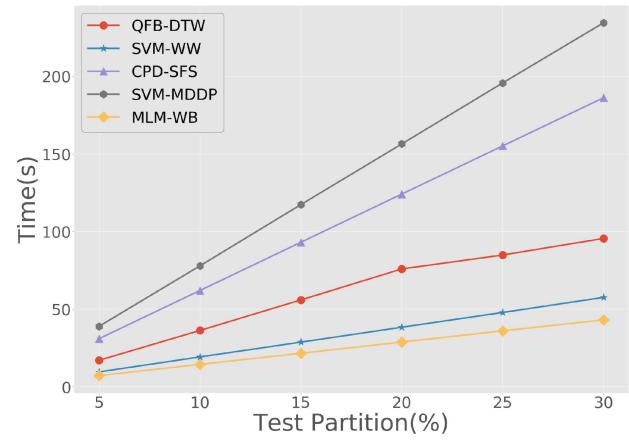


Fig. 12. Time consumption of data set 3.

the most time. The reason why method SVM-MDDP spends so much time is that this method calculates a large number of time domain and frequency domain features. Although this method performs well in the previous analysis, it takes too much resources to reach this performance. 3) Our method spends a medium degree of time. In data set 2, compared to the fastest method, our method will take 60% to 80% extra time consumption. However, the performance of those two fast methods are rather not ideal. Compared to the best

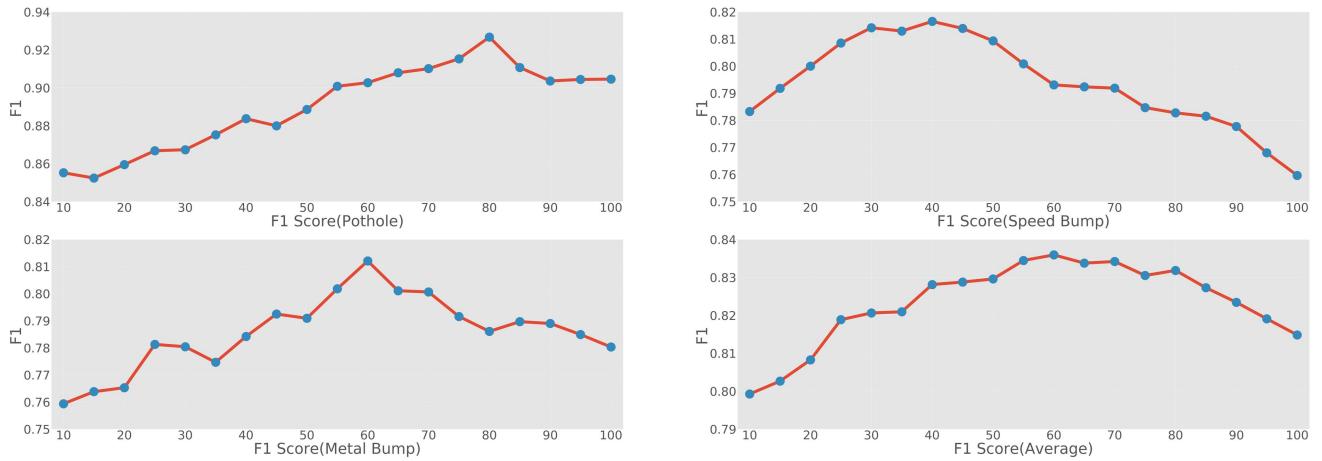


Fig. 13. Influence of velocity to our proposed method.

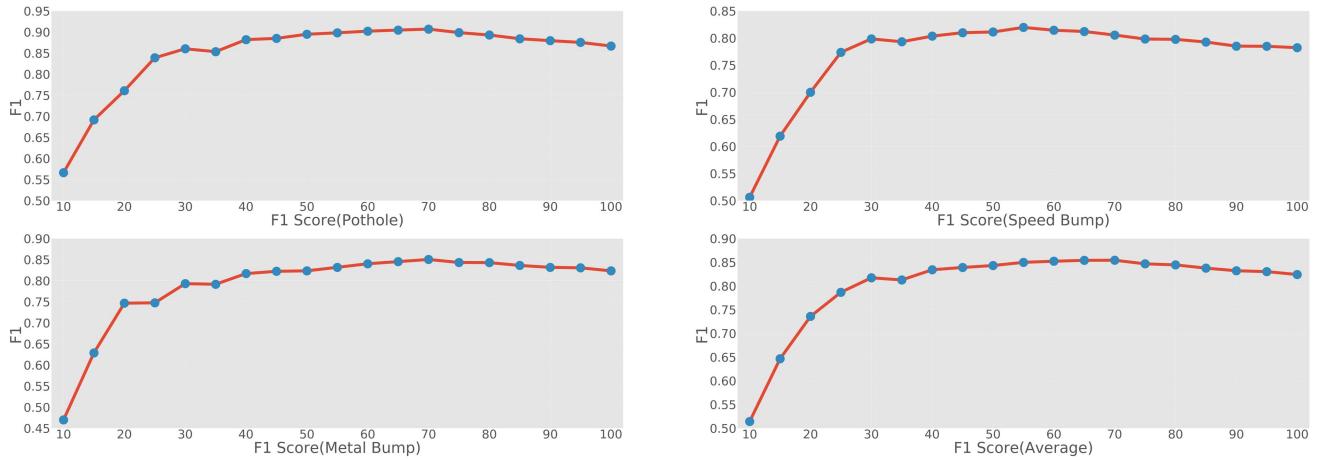


Fig. 14. Influence of sample rate to our proposed method.

performance method in the other methods, method SVM-MDDP, our method only take 1/8 time to perform a better result. In data set 3, method SVM-MDDP performs well, while its time consumption grows rapidly. Compared to those two methods that have a close performance, our method only takes 1/2 to 1/3 time to do calculation and prediction. Although we need extra 50% time consumption compared to the two methods spending the least time, we reach more than 60% better in performance.

In conclusion, we suppose we spend an acceptable more time consumption to reach a much better performance, or we spend much less time to reach a similar performance compared to the existing methods.

4) Performance With Different Velocity and Sample Frequency:

a) *Influence of different velocity:* As it can be seen in Figure 13, the f1 score first rises as the speed of the car increases, then start falling after a certain speed value. It can be explained that with the increase of the speed, the collision between the vehicle and the anomaly becomes more severe, causing more information in one anomaly window. The characteristics of anomalies become more apparent as the amount

of information increases. However, the time spent on the same length of anomaly will gradually become shorter, which means the anomaly window detected by the model decreases as the speed increases. A too short window makes a lot of information directly ignored, which reduces the recognition effect. For our model, f1 score reaches climax with the speed of 80, 40 and 60 respectively, and the average f1 score reaches the best with the speed of 60. This speed is close to the normal speed of the vehicle. As a result, our model can be applied to most vehicle in the case of edge computing, thus our model is fit for widely detecting road anomalies.

b) *Influence of different sample frequency:* As it can be seen in Figure 14, the f1 score first rises rapidly as the frequency increases, then start falling slowly after a certain frequency. It can be explained that in a low sample frequency, the data point increases rapidly from 10 to 20 and 20 to 30. As a result, there will be more information in one anomaly window. The characteristics of anomalies become more apparent as the amount of information increases. For our method, it will transform anomaly windows with different length into the same length, normally into the shortest length. The information losses during the change. With the

increase of sample frequency, the difference of the shortest length and the average length of window anomaly will increase as well, which causes more loss in information. Thus the f1 score decreases slowly over the sample frequency of 120. As a result, the sample rate in our model should be set in an appropriate value to avoid the loss.

V. CONCLUSION

Smart-phones are becoming easier to perform as data collection devices, making it more possible to analyze road conditions with these data. In this work we propose a model to detect the anomaly on the road based on a series of acceleration data. We first analyze the theoretical possibility of threshold detection and the random forest filter, then apply DTW to these time windows to define the anomaly type. We test our method in three data sets using F1 score as the metric to compare to some existing methods. Our method takes much advantage to all the other methods in three data sets, indicating that the model is usable in most situations. Our method fuses the advantage of machine learning methods and DTW algorithm, that is, to automatically identify the anomaly by machine learning methods, and to distinguish the type by DTW. We also avoid the influence of noise.

On the other hand, we do efforts to reduce the time consumption by random forest filter and adjusting the threshold. We take far less time consumption compared to the best existing method, which means we only use 1/7 to 1/8 of time consumption to reach a better performance.

In the future work, we would like to apply this system to a large amount of taxies and generate a map of the entire anomalies in the city with group perception. It requires the judgement and selection of the large amount of data and will help the government to repair the road in a more simple way.

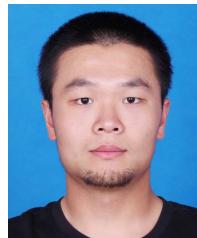
REFERENCES

- [1] J. Wahlstrom, I. Skog, and P. Handel, "Smartphone-based vehicle telematics: A ten-year anniversary," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2802–2825, Oct. 2017.
- [2] L. C. Gonzalez, R. Moreno, H. J. Escalante, F. Martinez, and M. R. Carlos, "Learning roadway surface disruption patterns using the bag of words representation," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 2916–2928, Nov. 2017.
- [3] U. M. Smartphones, P. Mohan, N. Venkata, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions," Microsoft, Redmond, WA, USA, Tech. Rep. MSRTR200859, 2008, pp. 323–336.
- [4] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: Using a mobile sensor network for road surface monitoring," in *Proc. 6th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*. New York, NY, USA: ACM, 2008, pp. 29–39, doi: 10.1145/1378600.1378605.
- [5] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, "Real time pothole detection using Android smartphones with accelerometers," in *Proc. Int. Conf. Distrib. Comput. Sensor Syst. Workshops (DCOSS)*, Jun. 2011, pp. 1–6.
- [6] M. Perttunen *et al.*, "Distributed road surface condition monitoring using mobile phones," in *Proc. Int. Conf. Ubiquitous Intell. Comput.* Berlin, Germany: Springer, 2011, pp. 64–78.
- [7] F. Seraj, B. J. van der Zwaag, A. Dilo, T. Luarasi, and P. Havinga, "Roads: A road pavement monitoring system for anomaly detection using smart phones," in *Big Data Analytics in the Social and Ubiquitous Context*, M. Atzmueller, A. Chin, F. Janssen, I. Schweizer, and C. Trattner, Eds. Cham, Switzerland: Springer, 2016, pp. 128–146.
- [8] M. R. Carlos, L. C. Gonzalez, F. Martinez, and R. Cornejo, "Evaluating reorientation strategies for accelerometer data from smartphones for its applications," in *Ubiquitous Computing and Ambient Intelligence*, C. R. Garcia, P. Caballero-Gil, M. Burmester, and A. Quesada-Arenzibia, Eds. Cham, Switzerland: Springer, 2016, pp. 407–418.
- [9] A. S. El-Wakeel, J. Li, A. Noureldin, H. S. Hassanein, and N. Zorba, "Towards a practical crowdsensing system for road surface conditions monitoring," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4672–4685, Dec. 2018.
- [10] F. Cong *et al.*, "Applying wavelet packet decomposition and one-class support vector machine on vehicle acceleration traces for road anomaly detection," in *Advances in Neural Networks—ISNN 2013*, C. Guo, Z.-G. Hou, and Z. Zeng, Eds. Berlin, Germany: Springer, 2013, pp. 291–299.
- [11] S. Hu, L. Su, H. Liu, H. Wang, and T. F. Abdelzaher, "Smartroad: Smartphone-based crowd sensing for traffic regulator detection and identification," *ACM Trans. Sen. Netw.*, vol. 11, no. 4, pp. 55:1–55:27, Jul. 2015. [Online]. Available: <http://doi.acm.org/10.1145/2770876>
- [12] A. Fox, B. V. K. V. Kumar, and F. Bai, "Multi-source variable-rate sampled signal reconstructions in vehicular CPS," in *Proc. IEEE INFOCOM-35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.
- [13] A. Fox, B. V. K. V. Kumar, J. Chen, and F. Bai, "Multi-lane pothole detection from crowdsourced undersampled vehicle sensor data," *IEEE Trans. Mobile Comput.*, vol. 16, no. 12, pp. 3417–3430, Dec. 2017.
- [14] Z. Li, D. P. Filev, I. Kolmanovsky, E. Atkins, and J. Lu, "A new clustering algorithm for processing GPS-based road anomaly reports with a Mahalanobis distance," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 7, pp. 1980–1988, Jul. 2017.
- [15] A. Allouch, A. Koubaa, T. Abbes, and A. Ammar, "RoadSense: Smartphone application to estimate road conditions using accelerometer and gyroscope," *IEEE Sensors J.*, vol. 17, no. 13, pp. 4231–4238, Jul. 2017.
- [16] T. S. Brisimi, C. G. Cassandras, C. Osgood, I. C. Paschalidis, and Y. Zhang, "Sensing and classifying roadway obstacles in smart cities: The street bump system," *IEEE Access*, vol. 4, pp. 1301–1312, 2016.
- [17] A. S. El-Wakeel, J. Li, M. T. Rahman, A. Noureldin, and H. S. Hassanein, "Monitoring road surface anomalies towards dynamic road mapping for future smart cities," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2017, pp. 828–832.
- [18] M. Jaworski, P. Duda, and L. Rutkowski, "New splitting criteria for decision trees in stationary data streams," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2516–2529, Jun. 2018.
- [19] Y. Ying and P. Li, "Distance metric learning with eigenvalue optimization," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 1–26, Jan. 2012. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2503308.2188386>
- [20] G. Xue, H. Zhu, Z. Hu, J. Yu, Y. Zhu, and Y. Luo, "Pothole in the dark: Perceiving pothole profiles with participatory urban vehicles," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1408–1419, May 2017.
- [21] G. Alessandroni, A. Carini, E. Lattanzi, V. Freschi, and A. Bogliolo, "A study on the influence of speed on road roughness sensing: The smartroadsense case," *Sensors*, vol. 17, no. 2, p. 305, Feb. 2017.
- [22] F. Seraj, P. J. M. Havinga, and N. Meratnia, "SpinSafe: An unsupervised smartphone-based wheelchair path monitoring system," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2016, pp. 1–6.
- [23] R. Madli, S. Hebbar, P. Pattar, and V. Golla, "Automatic detection and notification of potholes and humps on roads to aid drivers," *IEEE Sensors J.*, vol. 15, no. 8, pp. 4313–4318, Aug. 2015.
- [24] M. R. Carlos, M. E. Aragn, L. C. Gonzlez, H. J. Escalante, and F. Martnez, "Evaluation of detection approaches for road anomalies based on accelerometer readings—Addressing who's who," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 10, pp. 3334–3343, Oct. 2018.
- [25] P. M. Harikrishnan and V. P. Gopi, "Vehicle vibration signal processing for road surface monitoring," *IEEE Sensors J.*, vol. 17, no. 16, pp. 5192–5197, Aug. 2017.
- [26] C. W. Yi, Y. T. Chuang, and C. S. Nian, "Toward crowdsourcing-based road pavement monitoring by mobile sensing technologies," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1905–1917, Aug. 2015.
- [27] A. Pesterev, L. Bergman, and C. Tan, "Pothole-induced contact forces in a simple vehicle model," *J. Sound Vib.*, vol. 256, no. 3, pp. 565–572, 2002.
- [28] A. Allouch, A. Koubaa, T. Abbes, and A. Ammar, "Roadsense: Smartphone application to estimate road conditions using accelerometer and gyroscope," *IEEE Sensors J.*, vol. 17, no. 13, pp. 4231–4238, Jul. 2017.

- [29] A. S. El-Wakeel, A. Osman, A. Noureldin, and H. S. Hassanein, "Road test experiments and statistical analysis for real-time monitoring of road surface conditions," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.
- [30] A. Fox, B. V. K. V. Kumar, J. Chen, and F. Bai, "Crowdsourcing undersampled vehicular sensor data for pothole detection," in *Proc. 12th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2015, pp. 515–523.



Zengwei Zheng received the B.S. and M.Ed. degrees in computer science and western economics from Hangzhou University, China, in 1991 and 1998, respectively, and the Ph.D. degree in computer science and technology from Zhejiang University, China, in 2005. He is currently a Professor with the Department of Computer Science and Engineering, the Director of Intelligent Plant Factory of Zhejiang Province Engineering Laboratory, the Director of the Hangzhou Key Laboratory for IoT Technology and Application, and the Head of the Department of Scientific Research, Zhejiang University City College. His research interests include wireless sensor networks, location-based service, the Internet of Things, digital agriculture, and pervasive computing. He is a member of ACM and CCF.



Mingxuan Zhou received the B.S. degree in electrical engineering from Zhejiang University in 2017. He is currently pursuing the master's degree with the College of Computer Science and Technology, Zhejiang University and Zhejiang University City College. His research interests include the Internet of Things and ubiquitous computing.



Yuanyi Chen received the B.Sc. degree from Sichuan University in 2010, the M.Sc. degree from Zhejiang University in 2013, and the Ph.D. degree from Shanghai Jiao Tong University in 2017. He is currently a Distinguished Research Fellow with the Department of Computer Science and Computing, Zhejiang University City College. He has published more than 20 technical articles in major international journals and conference proceedings. His research interests include the Internet of Things, mobile computing, and ubiquitous computing.



Meimei Huo received the Bachelor of Medicine degree from the Taisan Medical College in 2000 and the master's degree in biomedical engineering from Zhejiang University in 2003. She is currently an Associate Professor with the Zhejiang University City College. Her research interests include the wireless sensor networks, the Internet of Things, and ubiquitous computing.



Lin Sun (Member, IEEE) received the B.S. degree in communication engineering and the M.S. degree in computer science from the East China University of Science and Technology, China, in 2001 and 2004, respectively, and the Ph.D. degree in computer science from Zhejiang University, China, in 2010. He is currently an Associate Professor with the Department of Computer Science, Zhejiang University City College. His research interests include pattern recognition and pervasive computing.



Sha Zhao (Member, IEEE) received the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2017. She is currently a Post-Doctoral Research Fellow with the College of Computer Science and Technology, Zhejiang University. She visited the Human-Computer Interaction Institute, Carnegie Mellon University, from September 2015 to September 2016. Her research interests include pervasive computing, mobile sensing, data mining, and machine learning. She received the Best Paper Award of ACM UbiComp 2016.



Dan Chen received the B.S. degree in electrical engineering and automation from Tianjin Polytechnic University in 1996 and the M.E. degree in computer Science from Zhejiang University in 2004. Since 2008, he has been an Associate Professor of computer science with the Zhejiang University City College. His research interests include computer networks, mobile computing, the IoT, and computer architecture.