Engineering Applications of Artificial Intelligence 74 (2018) 198–211

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Asphalt pavement classification using smartphone accelerometer and Complexity Invariant Distance

Vinicius M.A. Souza *

Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, SP, Brazil
Onion Tecnologia, São Carlos, SP, Brazil

ARTICLE INFO

Keywords:
Accelerometer
Smartphones
Asphalt pavement evaluation
Time series classification
Machine learning

ABSTRACT

Most modern smartphones have a variety of built-in sensors, as an accelerometer, gyroscope, GPS, proximity and a magnetic sensor. The large variety of sensors makes these devices powerful measurement tools, allowing the emergence of new systems and applications. In this paper, is presented a real data stream application related to asphalt pavement evaluation using acceleration data gathered by the accelerometer sensor of smartphones. The quality of the pavement has a significant influence on the final price of goods and services, the safety of drivers, pedestrians and passengers, and driver's comfort. Thus, it is essential the use of tools, as proposed in this work, that allows the constant monitoring of pavement conditions by the Government authorities or private entities for more precise interventions in the maintenance planning with fewer expenses. This task is mainly important in developing countries where there is a lack of technology and a reduced budget for maintenance. Due to the popularity of smartphones, this tool can make possible that different users help to monitor the pavement quality ubiquitously during driving periods without effort. The application of asphalt evaluation is modeled in this work as a multi-dimensional time series classification problem, where the time series are the data from the three-axis accelerometer sensor. Given the characteristics of the data, we discuss and propose the combination of some classical distance measure for time series as Dynamic Time Warping or Longest Common Subsequence Similarity with the Complexity Invariant Distance. A comprehensive experimental evaluation was performed on three datasets that represent different scenarios of asphalt pavement classification. The proposed approach reaches a classification accuracy of 80% to 98% in the three evaluated problems and we experimentally show that the complexity invariance substantially improves the results achieved by classical distance measures given our classification tasks.

1. Introduction

In the last years, the number of smartphone users has rapidly increased. In 2007, the number of users worldwide was 400 million.[1] In 2016, smartphone sales to end users totaled nearly 1.5 billion units, an increase of 5 percent from 2015, according to Gartner, Inc.[2] It is expected that this number to pass the 5 billion mark by 2019.[3]

Most modern smartphones have a variety of embedded sensors, as an accelerometer, gyroscope, Global Positioning System (GPS), proximity and a magnetic sensor. These sensors allow smartphones to be used as useful measurement tools to collect a large amount of data from users and environmental. Consequently, emerge different sensing applications by portable devices as handwriting, gesture, and activity recognition (Kim et al., 2014; Liu et al., 2009; Ronao and Cho, 2016), road traffic detection (Predic and Stojanovic, 2015), among others.

In this direction, the industries of vehicles and navigation systems has gained new ways to collect data, which in turn has come to benefit drivers, vehicle owners, and society as a whole. According to Engelbrecht et al. (2015), the existing literature on vehicle sensing based on smartphones can be categorized according to the four types of

* Correspondence to: Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, SP, Brazil.
E-mail address: vinicius@oniontecnologia.com.br.

[1] http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/.

[2] http://www.gartner.com/newsroom/id/3609817.

[3] https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/.

https://doi.org/10.1016/j.engappai.2018.06.003
Received 7 June 2017; Received in revised form 3 November 2017; Accepted 12 June 2018
Available online 6 July 2018

information that are captured: (*i*) traffic; (*ii*) vehicle; (*iii*) environmental; and (*iv*) driver behavior. This information is then disseminated in various ways for different applications.

In this paper, we have a particular interest to collect the environmental information using portable devices. Specifically, information about asphalt pavement of roads and streets given by the accelerometer sensor while the user driving a vehicle.

In most of the countries, streets and roads play a key role in the transportation of cargo, products, and passengers by means of vehicles. The quality of the pavement has significant influence on the final price of goods and services, the safety of drivers, pedestrians and passengers, and driver's comfort. Thus, it is essential the use of tools that allow the constant monitoring of pavement conditions by the Government authorities or private entities for more precise interventions in the maintenance planning with fewer expenses. This task is mainly important in developing countries where there is a lack of technology and reduced budget for maintenance.

In order to reduce the effort and time cost of manual inspections made by experts or the need of use of high-cost equipment as laser profilometer, smartphones can be used to evaluate the pavement of roads and streets. Due to the popularity of smartphones, this tool can make possible that different users help to monitor the pavement quality ubiquitously during driving periods without effort.

In this paper, the task of asphalt pavement evaluation is divided into three different classification problems. In the first scenario, the goal of the classification is the identification of regular or deteriorated pavement asphalt. In the second scenario, the classification task is the identification of pavement type as cobblestone or dirt road. In the last scenario, the classification task is the identification of different obstacles in the road, as the speed bumps and raised markers.

Our problem is modeled in this paper as a time series classification task, where the time series are the data from the accelerometer sensor. As suggested by the literature, a simple One-Nearest Neighbor algorithm (1NN), with a proper distance function, presents very good results, frequently outperforming more complex classification algorithms (Ding et al., 2008). However, the choice of an adequate distance measure is essential and depends on the problem domain. Given the characteristics of the data, we discuss and propose the combination of some classical distance measure with the Complexity Invariant Distance (CID) (Batista et al., 2014).

As most of the smartphones have sensors that can measure the acceleration force on three axes (x, y, z), we have a multi-dimensional time series classification problem. We propose enrich our classifier combining the information of each acceleration axis by means of confidence levels provided by Conformal Prediction (Vovk et al., 2005) and different strategies to combine the data from different axes as majority vote and sum rule. To the best of our knowledge, the use of Conformal Prediction to combine classifiers has never been used in the literature for multi-dimensional time series classification.

Four popular distance measures for time series was evaluated alone and combined with CID: (*i*) Dynamic Time Warping (DTW), (*ii*) Longest Common Subsequence Similarity (LCSS), (*iii*) Derivative Dynamic Time Warping (DD_{DTW}), and (*iv*) Derivative Transform Distance (DTD_{DTW}). Although CID allows their use with any distance measure, the works from literature never used distance measures as LCSS, DD_{DTW}, and DTD_{DTW} with a complexity distance.

In summary, among different configurations to perform the classification task of asphalt pavement using data gathered by an accelerometer, we can suggest the use of a 1NN classifier with the distance measure CID-LCSS or CID-DD_{DTW}. With these configurations, it is possible to achieve a classification accuracy of 80% to 98% in the three classification problems evaluated in this paper.

The main contributions of this paper are the proposal of a simple and low-cost system to perform the evaluation of asphalt pavement, the collection and built of three different datasets, the proposal of Conformal Prediction for multi-dimensional time series classification, the evaluation of different distance measures for time series to choose the most adequate for the problem, and the confirmation of the importance of a complexity invariant distance to improve the results of classical distance measures.

The remainder of this paper is organized as follows. In Section 2, related works that use data from smartphones to evaluate the asphalt pavement are presented. In Section 3 is presented the data discussed and evaluated in this paper, as well as the procedures of data collection and preprocessing. Section 4 provides a background related to time series classification and distance measures. In Section 5 is presented a precise way to determine confidence levels in classification problems called Conformal Prediction. The proposed approach is presented in Section 6. The experimental evaluation is presented in Section 7. Lastly, our conclusions and future works are presented in Section 8.

2. Related work

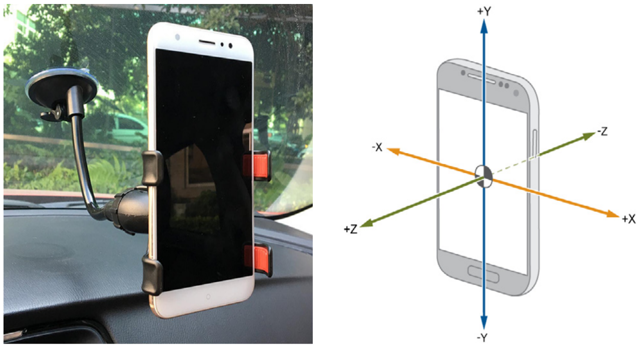
Most of works has been devoted only to detect and report potholes. In general, these papers are based on the use of thresholds or simple analysis of the series, as the distance between two peaks. In this sense, we can consider that this work is the first one to deal with the problem of pavement evaluation as time series classification and it shows a more comprehensive experimental evaluation, never considered by other works. The works closest to our are: Pothole Patrol (Eriksson et al., 2008), Nericell (Mohan et al., 2008), the system proposed by Mednis et al. (2011), and Street Bump (Brisimi et al., 2016).

In Pothole Patrol (Eriksson et al., 2008), the three-axis acceleration data is gathered at high frequency (380 times per second) and the main goal of the system is to detect potholes. Signal processing filters are applied to reject one or more non-pothole event types as a manhole, expansion joint, and railroad crossing. The filters are based on the analysis of speed (to reject door slams and curb ramps), peaks of acceleration in the z-axis, and peaks of acceleration in the z-axis when the acceleration in the x-axis is below a threshold value. The authors report an accuracy of 92% for pothole identification

Nericell (Mohan et al., 2008) is a system to monitor road and traffic conditions. It detects potholes, braking, speed bumps, and honks using an accelerometer, microphone, GSM radio and GPS sensors of smartphones. For potholes, the detection is based on the analysis of z-values of the accelerometer. It also provides two heuristics based upon the speed of the vehicle. If speed is greater than 25 km/h, it uses z-peak heuristic where a spike along z-value above a specific threshold is classified as a pothole. At low speed, the z-sus heuristic is used which detects a sustained dip in z-value for at least 20 ms. The false negative rate for both detectors is high (20%–30%) for speed bump detection.

The system proposed by Mednis et al. (2011) uses four algorithms to detect potholes: Z-THRESH, Z-DIFF, STDEV(Z) and G-ZERO. Z-THRESH identifies the type of pothole (small pothole, a cluster of potholes, large potholes) based on the values of acceleration observed in the z-axis above a specific threshold level. In Z-DIFF, is performed a search in the acceleration values of z-axis for two consecutive measurements with a difference between their values above a specific threshold level. STDEV(Z) algorithm calculates the standard deviation of accelerometer data in the vertical direction over a specified window size. G-ZERO searches for the case where all the three-axis data values are near to 0g. This measure indicates that the vehicle it is in a temporary free fall. The best result is achieved by Z-DIFF algorithm, with 92% of true positives to the identification of pothole type.

Street Bump (Brisimi et al., 2016) is a system capable of classifying roadway obstacles into predefined categories using machine learning algorithms. They use the term ”bump” in a generic sense to describe various obstacles which include potholes, sunk castings (manhole covers), utility patches, catch basins (drains), train tracks and speed bumps. In Brisimi et al. (2016) is presented an evaluation of supervised learning algorithms as SVM, AdaBoost, logistic regression, and random forests and unsupervised learning inspired by anomaly detection problems. The system extracts two main features of the acceleration data, the first



(a) Phone holder

(b) Acceleration axes

Fig. 1. Example of car phone holder used to collect data and directions considered by the accelerometer sensor.

is based on a Mean Squared Error (MSE) measure of a bump signal's deviation from that of a simple harmonic oscillation. The second feature relies on the entropy of a bump signal. In a dataset with 813 bumps collected in the City of Boston, they reported an accuracy of 88% to detection bumps.

3. Asphalt temporal data

In this section, the data used in the experimental evaluation is presented, as well as the procedures for data collection and preprocessing. As this data is generated by smartphone sensors (accelerometer and GPS) during the driving of a car over time and are responsible for measuring the asphalt conditions, they are called *asphalt temporal data*.

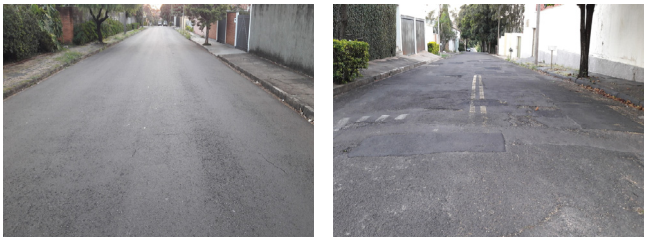
3.1. Data collection

In order to collect the data, a smartphone was installed inside the vehicle using a flexible suction holder near the dashboard, as shown in Fig. 1-(a). An expert is responsible for driving the vehicle while the device runs an Android application called *Asfault* (Souza et al., 2017), developed specifically to store the current asphalt condition continuously over time. This application is currently under development and it will be incorporated in a commercial tool in the future. *Asfault* stores the time-stamp of the collected data, acceleration forces in m/s^2 along the three physical axes (x, y, z) as shown in Fig. 1-(b), latitude, longitude, and velocity.

The acceleration forces are given by the accelerometer sensor of the device and are the data used for the classification task. Latitude, longitude, and velocity are given by the GPS and are used to locate the collected/classified data. Each one of this information is a different continuous time series. A time series can be defined as a sequence of n real numbers obtained through repeated measurements over time. A sampling rate of 100 Hz, which means 100 observations per second for each time series, was considered in this paper. This setting was used because is reachable for most of the devices. However, it is important to note that the Android system does not guarantee a perfect precision on the interval between readings. This means that when we choose a sampling rate of 100 Hz, the device should outputs between 95 and 105 observations per second.

In order to obtain labeled data, the *Asfault* application allows the expert to inform the pavement condition before collecting the data. To guarantee the integrity of data, *Asfault* also records videos of the road over the data collection using the built-in camera. Thus, it is possible to perform the analysis of these videos to confirm the class labels assigned by the expert.

To better perform the experiments and organize the discussion of results, the data was divided into different problems and was built three datasets. These datasets are introduced in the following sections.



(a) Regular asphalt

(b) Deteriorated pavement

Fig. 2. Photos that illustrate the two classes of the Asphalt-Regularity dataset.

Table 1
Class distribution and characteristic of the time series from Asphalt-Regularity dataset.

Class	Series length			Examples	Distribution
	Min	Max	Mean		
Regular	66	2371	238	762	50.73%
Deteriorated	190	4201	534	740	49.27%

The datasets were collected in the Brazilian cities of São Carlos, Ribeirão Preto, Araraquara, and Maringá using a medium sized hatchback car (Hyundai i30) and two different devices (Samsung Galaxy A5 and Samsung S7). Both devices are not used at the same in the data collection, but each one was used in different driving periods of data collection, as discussed in the following sections.

3.2. Asphalt pavement regularity

Initially, a problem with two classes based on the comfort felt by the driver according to the condition of the pavement was considered. This dataset has been named Asphalt-Regularity, and it considers the following two classes:

- *Regular*: when the pavement is regular and the driver comfort is very little changed over time;
- *Deteriorated*: when is observed some irregularities and roughness in a deteriorated pavement that are responsible for transferring vibrations to the cabin of the vehicle, reducing the comfort of the driver.

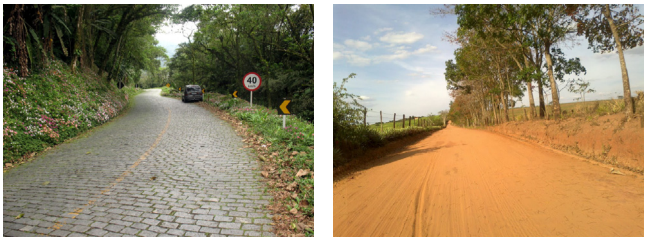
To better understand the problem, the classes of this dataset are illustrated in Fig. 2

For the Asphalt-Regularity dataset, were collected 1502 examples. The examples distribution over the class labels and some characteristics of the time series as the minimum, mean, and maximum length (in terms of the number of observations) are presented in Table 1. All the examples of this dataset were collected using a device model Samsung Galaxy A5.

Although this dataset is quite simple, it was used for an initial proof of concept for the use of the accelerometer for the task of evaluating pavements. The preliminary experiments conducted in this data were essential to deal with more complex problems, as the other two datasets.

3.3. Asphalt pavement type

In the second problem addressed in this work, the classification task of different pavement types was performed. Three class labels was considered in this data: (i) flexible pavement, (ii) cobblestone streets, and (iii) dirt roads. Flexible pavement can be defined as the one consisting of a mixture of asphaltic or bituminous material and aggregates placed on a bed of compacted granular material of appropriate quality in layers over the subgrade. Flexible pavements are preferred over cement concrete roads because they can be strengthened and improved in stages with



(a) Cobblestone street (b) Dirt road

Fig. 3. Photos that illustrate the class labels *Cobblestone street* and *Dirt road* of the Asphalt-PavType dataset.

Table 2
Class distribution and characteristic of the time series from Asphalt-PavType dataset.

Class	Series length			Examples	Distribution
	Min	Max	Mean		
Flexible	66	2371	246	816	38.65%
Cobblestone	284	1543	518	527	24.97%
Dirt road	274	1045	484	768	36.38%

Table 3
Class distribution and characteristic of the time series from Asphalt-Obstacles dataset.

Class	Series length			Examples	Distribution
	Min	Max	Mean		
Speed bump	178	730	330	212	27.14%
Vertical patch	114	279	191	222	28.43%
Raised markers	111	462	256	187	23.94%
Raised crosswalk	258	736	457	160	20.49%

the growth of traffic (Mohod and Kadam, 2016). An example of flexible pavement was previously shown in Fig. 2. However, for the Asphalt-PavType dataset, the flexible pavements with deteriorated conditions were not considered. The other two class labels of the dataset are illustrated in Fig. 3.

For the Asphalt-PavType dataset, were collected 2111 examples. The examples distribution over the three class labels is shown in Table 2. For this dataset, the first half of data was collected using the device model Samsung Galaxy A5 and another half was collected using the device model Samsung Galaxy S7.

This dataset can allow that the users of *Asfault* system choose more adequate routes with only paved roads, contributing to their safety and reducing the travel time.

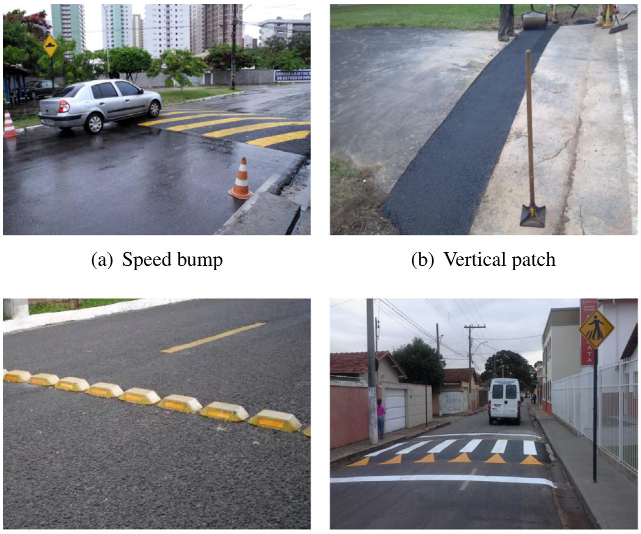
3.4. Asphalt obstacles

The last dataset built in this work consider the identification of four common obstacles in the region of data collection. This dataset has been named as Asphalt-Obstacles and has the following class labels: (i) speed bump, (ii) vertical patch, (iii) raised pavement markers, and (iv) raised crosswalk.

Examples from the Asphalt-Obstacles dataset are shown in Fig. 4. Although this dataset has four class labels, there is a great variability of the examples of the same class, given the lack of pattern in the build of these obstacles concerning width, height, size, and material.

For the Asphalt-Obstacles dataset, were collected 781 examples. The examples distribution over the four class labels of the problem is shown in Table 3. All the examples of this dataset were collected using a device model Samsung Galaxy S7.

In Fig. 5, is shown the location on the map of most of the examples belonging to the Asphalt-Obstacles dataset, collected at São Carlos city in Brazil.



(a) Speed bump (b) Vertical patch

(c) Raised pavement markers (d) Raised crosswalk

Fig. 4. Photos that illustrate the four class labels of the Asphalt-Obstacles dataset.

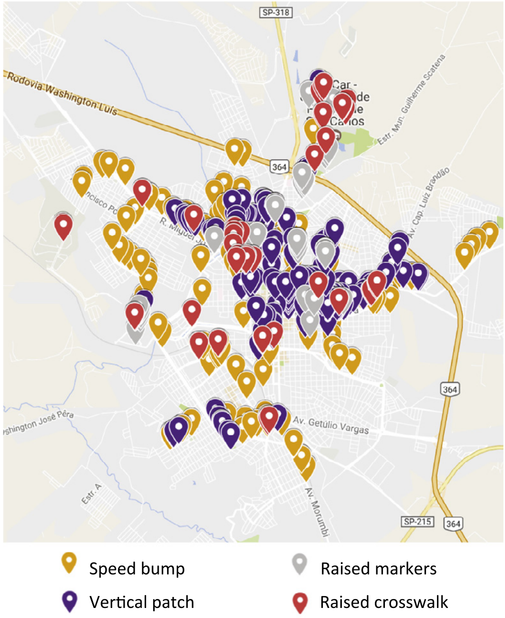


Fig. 5. Location of examples from Asphalt-Obstacles dataset collected at São Carlos city.

3.5. Preprocessing phase

A three-axis accelerometer detects the acceleration forces A in three perpendicular directions (x, y, z), resulting in the time series A_x, A_y, A_z. Thus, an accelerometer at rest on the surface of the Earth with an inclination of 90°, will measure a continuous acceleration $A_y \approx 9.81\ m/s^2$, given the gravity force. If the inclination of the sensor is changed for any direction, this acceleration force is distributed along the axes. In order to make the accelerometer data invariant to the inclination of the device, the force of gravity is removed in a preprocessing step by a high-pass frequency filter. The series of each axis were filtered with a $4th$ order Butterworth filter with a cutoff frequency $\omega = 0.25$ Hz (Selesnick and Burrus, 1998).

Table 4
Matlab code to calculate the acceleration magnitude and gravity removal.

```
function Am = accelMagnitude(Ax, Ay, Az)
    mag = sqrt(sum(Ax.^2 + Ay.^2 + Az.^2, 2));
    Am = mag - mean(mag);
```

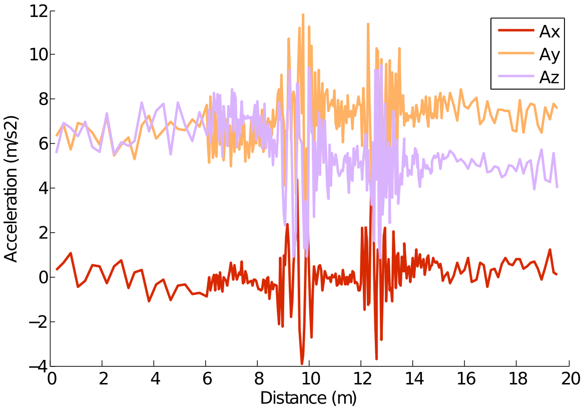


Fig. 6. An example of acceleration data obtained after crossing a raised pavement markers obstacle.

A more simple procedure for gravity removal was also considered. In this case, the time series A_x, A_y, A_z are converted in a unique time series that represents the acceleration magnitude A_m. The entire Matlab code to calculate the acceleration magnitude without the gravity force is presented in Table 4.

In order to illustrate these preprocessing steps, consider the example data obtained after crossing a raised pavement markers from the Asphalt-Obstacles dataset, as shown in Fig. 6. The resulting data after the gravity force removal by the high-pass filter is shown in Fig. 7-(a) and the acceleration magnitude A_m is shown in Fig. 7-(b).

For the datasets Asphalt-Regularity and Asphalt-PavType, a procedure of data segmentation or windowing was performed. In these datasets, the data was collected for more than 100 km (approximately 45 km for Asphalt Regularity and 63 km for Asphalt PavType) and the split of these data in minor segments (defined in meters by the user) is necessary to perform the evaluation procedures and to better indicate the location of asphalt problems on a map. A length of approximately 30 m for each segment data was considered in our experiments.

In Fig. 8, the segmentation procedure is illustrated given an example of acceleration data (A_m) collected on a cobblestone street. In this example, the time series has 9000 observations and approximately 600 m. Thus, this data will be split in 20 minor segments, generating 20 examples for the dataset.

In order to find the segments, the values of velocity (in m/s^2) of each observation divided by the current sampling rate are summed until this value reaches approximately 30 m. This procedure is because the sampling rate obtained by the smartphones used in the experiments varies from 95 to 105 Hz, as previously discussed. Thus, each segment of data represented by a time series has a different number of observations according to the velocity and sampling rate observed during the data collection. For example, if the data is collected in a constant velocity of 40 km/h (or 11.11 m/s) and a sampling rate of 100 Hz, a time series with approximately 30 m will have 270 observations. If the sampling rate is reduced to 95 Hz, the number of observations of the time series will be 256. However, in practice, the velocity and sampling rate are not constant over time.

4. Time series classification

A time series $T = (t_1, t_2, \dots, t_n)$ is an ordered set of n real values. The total number of real values is equal to the length of the time series. For clarity, each value t_i is refer as a *observation* measured in a given time i. A dataset $D = \{T_1, T_2, \dots, T_M\}$ is a collection of M time series.

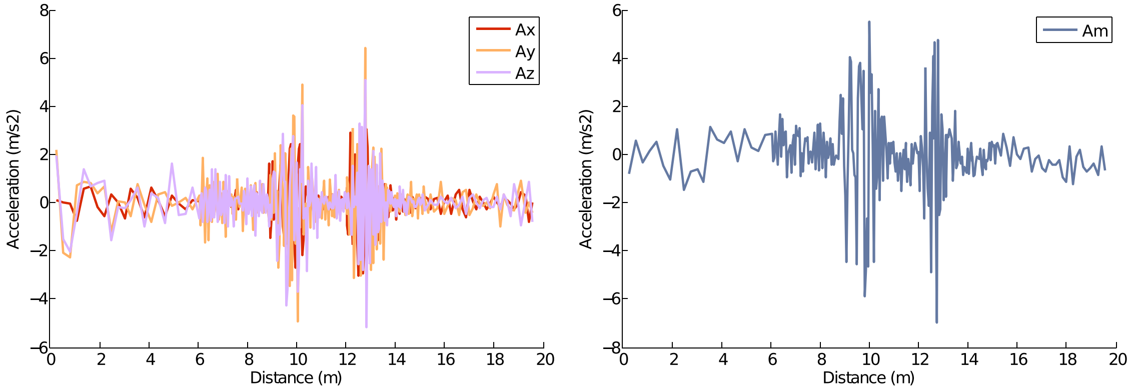
A time series is often the result of the observation of an underlying process which values are collected from measurements made at uniformly spaces time instants and according to a given sampling rate. In this paper, the time series data are the acceleration observed in the axes x, y, and z, measured by the accelerometer of a smartphone over driving periods of a vehicle with a sampling rate of approximately 100 Hz.

In classification, we are interested in assigning a class label c from a set of predefined class to an unknown query time series $Q = (q_1, q_2, \dots, q_n)$. Literature indicates that a simple One-Nearest Neighbor (1NN) algorithm, with a proper distance function, presents very good results, frequently outperforming more complex classification algorithms (Ding et al., 2008). Also, 1NN is parameter-free, unlike other classifiers that have several parameters to tune.

The 1NN algorithm assigns to the unlabeled query time series Q the label c_i of the most similar time series T_i from a labeled dataset $L = \{(T_1, c_1), \dots, (T_M, c_M)\}$ according to a distance measure $\mathcal{D}$. The distance measure $\mathcal{D}(T, Q)$ between time series T and Q is a function taking two time series as inputs and returning the similarity between these series.

The 1NN algorithm leaves open the choice of a distance measure. A simple and widely used measure is the Euclidean distance (ED), as defined in Eq. (1).

$$ED(T, Q) = \sqrt{\sum_{i=1}^{n} (t_i - q_i)^2} \tag{1}$$



(a) Preprocessed data

(b) Acceleration magnitude A_m

Fig. 7. Example of acceleration data obtained crossing a raised pavement markers, after the removal of gravity force by a high-pass filter *(a)* and the acceleration magnitude A_m *(b)*.

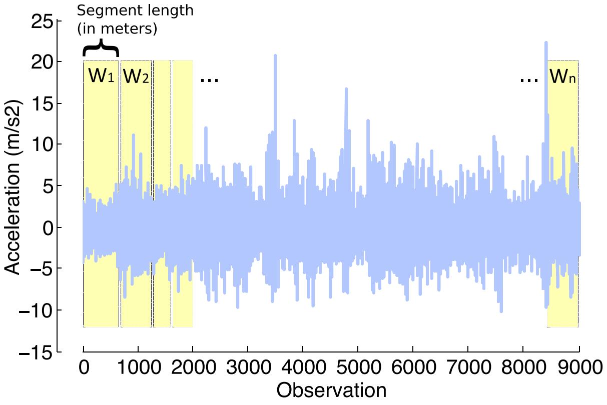


Fig. 8. Procedure of data segmentation or windowing given an example of acceleration data collected in a cobblestone street. The segment length in terms of number of observations varies according to the sampling rate and velocity.

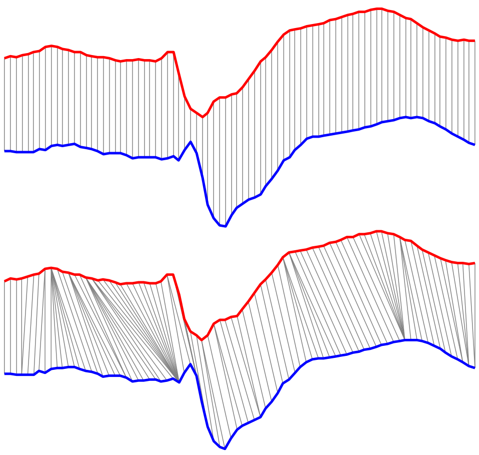


Fig. 9. Difference between the alignments achieved by Euclidean distance (*top*) and DTW (*bottom*) (Silva et al., 2013).

The Euclidean distance measures the similarity of two time series comparing the observations in the exact same moment i. However, many applications require a more flexible matching, in which an observation t_i at time i can be associated to an observation q_j at time j, where $i \neq j$. Also, Euclidean distance requires that both time series have exactly the same length n. In many applications, such as the one explored in this work, this requirement cannot be fulfilled without a resample procedure, being necessary to use more appropriate measures. The main distances from literature that allows *elastic* shifting of the time axis are discussed in this section.

4.1. Dynamic Time Warping

The Dynamic Time Warping (DTW) distance achieves an optimal non-linear alignment of the observations under some constraints. Fig. 9 illustrates the difference between the linear alignment obtained by the Euclidean distance, and the non-linear alignment obtained by the DTW algorithm.

To align two time series T and Q using DTW, an n-by-m matrix is constructed, where n is the length of T and m is the length of Q, respectively. In this matrix, the (ith, jth) element is the squared Euclidean distance $ED(t_i, q_j)$ between the observations t_i and q_j. The non-linear matching of the series is given by a warping path. A warping path $P = (p_1, p_2, \ldots, p_T)$ is a contiguous set of matrix elements under the following constraints:

- Boundary constraint. The matching is made for the whole time series T and Q. Therefore, it starts at the pair $(1, 1)$ and ends at (n, m);
- Continuity constraint. The matchings are made in one-unit steps. It means that the matching never "jumps" one or more observations of any time series;
- Monotonicity constraint. The relative order of observations has to be preserved, i.e., if $s1 < s2$, the matching of t_i with q_{s1} is done before matching t_i with q_{s2}.

While there are exponentially many warping paths that satisfy the above conditions, we are only interested in the path that minimizes the warping cost. Thus, the final value of similarity between the time series achieved by DTW is given by Eq. (2).

$$DTW(T, Q) = min \left\{ \sqrt{\sum_{t=1}^{T} p_t} \right\} \tag{2}$$

This path can be found using dynamic programming. The algorithm is based on the initial condition described in Eq. (3).

$$DTW(i, j) = \begin{cases} 0, \text{ if } i = j = 0 \\ \infty, \text{ if } i = 0 \text{ or if } j = 0 \end{cases} \tag{3}$$

The recurrence relation of DTW algorithm is presented in Eq. (4).

$$DTW(i, j) = c(t_i, q_j) + min \begin{cases} DTW(i-1, j) \\ DTW(i, j-1) \\ DTW(i-1, j-1) \end{cases} \tag{4}$$

where $i = 1 \ldots n$ and $j = 1 \ldots m$ and n and m are the lengths of the T and Q time series, respectively. $c(t_i, q_j)$ is the cost of matching two observations t_i and q_j, frequently calculated with Euclidean distance. The resulting value in $DTW(n, m)$ is the DTW distance between T and Q.

In order to improve the efficiency of DTW calculations, the use of warping windows is common. Warping window, or constraint band, defines the maximum allowed time difference between two matched observations. A typical constraint is the Sakoe–Chiba Band which states that the warping path cannot deviate more than R cells from the diagonal (Sakoe and Chiba, 1978).

4.2. Derivative Dynamic Time Warping

Some distance measures are based on the first order differences of the series. The first derivative T' of a time series T with length n can be found by the differences between adjacent elements. Thus, the observation i of T' is given by $T'(i) = t_i - t_{i+1}$ for $i = 1$ to $n - 1$.

The idea beyond the use of derivative of the series is to prevent unintuitive alignments made by DTW where a single point on one time series maps onto a large subsection of another time series. The examples of this undesirable behavior are called "singularities". This problem occurs because DTW may try to explain variability in the y-axis by warping the x-axis or when a feature (i.e. peak, valley, inflection point, plateau, etc.) in one sequence is slightly higher or lower than its corresponding feature in the other sequence (Keogh and Pazzani, 2001).

Consider the two artificial time series shown in Fig. 10-(1). In this example, we have some points in one of the series where we have a rising trend, while the other series shows a falling trend. In Fig. 10-(2), we show four expected intuitive local warping. However, classic DTW returns four different singularities, as shown in Fig. 10-(3). When we consider the derivative of the time series, we can prevent these singularities and obtain a better alignment as shown in Fig. 10-(4).

Methods as described by Jeong et al. (2011), use just the derivatives of the series to measure their similarity. However, most successful approaches combine distance in the time domain and the difference domain (Bagnall et al., 2016).

Górecki and Łuczak (2013) proposed the Derivative Dynamic Time Warping (DD_{DTW}), an approach for using a weighted combination of

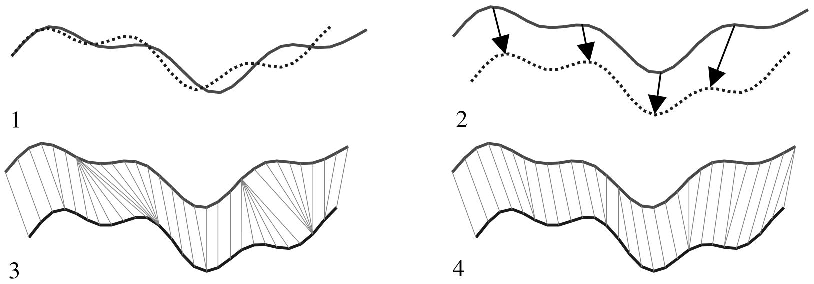


Fig. 10. (1) Two artificial time series examples; (2) Four intuitive local points that are expected warping alignment; (3) The alignment achieved by classic DTW; (4) The alignment achieved when is considered the derivative of both series, without singularities (Keogh and Pazzani, 2001).

raw series and first-order differences for 1NN classification. They find the DTW distance between two series and the two differenced series. These two distances are then combined using a weighting parameter α. The authors suggest the search for values for α in the training data through a leave-one-out cross-validation.

Given two time series T and Q, and their respective derivatives T' and Q', the final distance of $\mathrm{DD_{DTW}}$ between T and Q is found according to Eq. (5), where $DTW(T,Q)$ is the distance achieved by the Dynamic Time Warping.

$$DD_{DTW}(T,Q) = \alpha \times DTW(T,Q) + (1-\alpha) \times DTW(T',Q') \tag{5}$$

4.3. Derivative Transform Distance

The Derivative Transform Distance ($\mathrm{DTD_{DTW}}$) (Górecki and Łuczak, 2014) is an extension of $\mathrm{DD_{DTW}}$ that uses DTW in conjunction with transforms and derivatives. The authors suggest the use of functions as sine, cosine, and Hilbert to transform the series. For example, given the series Q with length n, the transformed series Q'' by the cosine function is achieved by Eq. (6).

$$Q'' = \sum_{i=1}^{n} Q_i \, cos\left[\frac{\pi}{n}\left(i-\frac{1}{2}\right)(i-1)\right] \tag{6}$$

The combination of derivatives and transforms considered by $\mathrm{DTD_{DTW}}$ is given by Eq. (7), where Q' is the derivative and Q'' is the transform of Q.

$$DTD_{DTW}(T,Q) = \alpha \times DTW(T,Q) + \beta \times DTW(T',Q') + (1-\alpha-\beta) \times DTW(T'',Q'') \tag{7}$$

The parameters $\alpha, \beta \in [0,1]$ are chosen in the learning phase. The authors suggest the use of cross-validation for tuning these parameters in the training set.

4.4. Longest Common Subsequence Similarity

The Longest Common Subsequence (LCS) is the problem of finding the longest subsequence common to all sequences in a set of sequences. This problem has been studied for several decades to measure the similarity between two strings (Hirschberg, 1975).

Consider two strings r_1 and r_2. By deleting the characters from a sequence without changing the order of the remaining characters, we can get a subsequence. The LCS is the common subsequence of both r_1 and r_2 with the maximum length. Suppose that the string r_1 is composed by the following characters $r_1 = ABAZDC$ and $r_2 = BACBAD$. In this example, the LCS has length 4 and is the string $ABAD$. Another way to look for this problem is to find a 1–1 matching between letters of r_1 in r_2 such that none of the edges in the matching cross each other.

The measure was adapted to deal with the alignment of time series and then called of Longest Common Subsequence Similarity (LCSS) (Vlachos et al., 2003). Given a query and a target time series, LCSS finds subsequences of the query and target (of the same length) that best correspond to each other. The distance is based on the ratio between the length of longest common subsequence and the length of the whole sequence. The subsequence does not need to consist of consecutive points, the order of points is not rearranged, and some points can remain unmatched (Latecki et al., 2005).

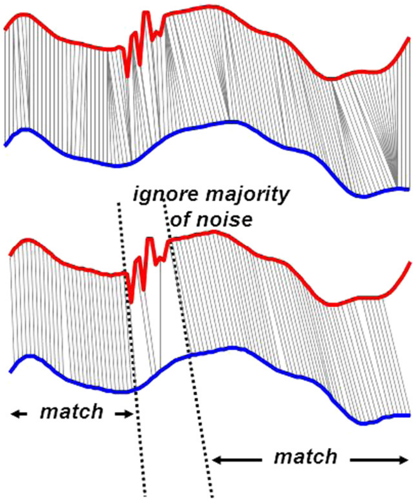


Fig. 11. Difference between the alignments achieved by Dynamic Time Warping (*top*) and Longest Common Subsequence Similarity (*bottom*) (Vlachos et al., 2003).

The traditional LCS algorithm considers only two states of a character pair, *match* and *mismatch*. However, time series are composed of real numbers. Thus, the data point pair of real numbers cannot be determined to fall into the only two states of *match* and *mismatch*. The distance between two data points of real numbers is used to determine whether they are close enough to be regarded as a *match* or not. More specifically, a threshold parameter ϵ was introduced, stating that two points from two time series are considered to match if their distance is less than ϵ (Vlachos et al., 2003).

When compared with DTW, the main advantage of LCSS is their ability to deal with the presence of outliers or noise in the series, given that the measure can ignore some data in the alignment. In Fig. 11 we shown a comparison of the alignments achieved by DTW and LCSS, given two examples of time series. Although both series shows a similar shape, the time series in red (top) has some noisy observations that can be ignored in the alignment.

4.5. Complexity Invariant Distance

The success of a distance function to measure the similarity of two time series in a mining task is associated to its capacity to correctly deal with the invariances required by the domain. For example, if we try to

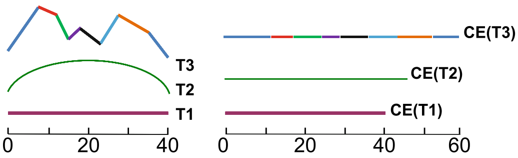


Fig. 12. Three time series (*left*) can have their complexity measured by stretching them and measuring the length of the resulting lines (*right*) (Batista et al., 2014). For the time series T3, 8 resulting lines are achieved and represented in different colors. Specifically, each segment starts at the positions 0, 10, 18, 25, 30, 38, 45, and 50, respectively, given the y-axis from the right.

compare two time series measured on different scales, say Celsius and Fahrenheit, they will not match well, even if they have similar shapes. To measure the true underlying similarity, we must first make their amplitudes the same (invariance to amplitude). Data from biological signals, motion capture, handwriting, and ECGs require invariance to warping, and cardiology data requires invariance to mean value.

Batista et al. (2014) noted that pairs of complex objects, even those which subjectively may seem very similar to the human eye, tend to be further apart under current distance measures than pairs of simple objects. This fact introduces errors in nearest neighbor classification because complex objects are incorrectly assigned to a simpler class. To deal with this problem, Batista et al. (2014) proposed the Complexity Invariant Distance (CID) for time series.

Complexity invariance uses information about complexity differences between two time series as a correction factor for existing distance measures. Considering the Euclidean distance, $ED(T,Q)$, between two time series T and Q, the distance can be made complexity-invariant by introducing a correction factor according to Eq. (8).

$$CID(T,Q) = ED(T,Q) \times CF(T,Q), \tag{8}$$

where CF is a complexity correction factor defined according to Eq. (9):

$$CF(T,Q) = \frac{max(CE(T), CE(Q))}{min(CE(T), CE(Q))}, \tag{9}$$

and $CE(T)$ is a complexity estimated of a time series T. The correction factor CF forces time series with very different complexities to be further apart. When all time series have the same complexity, CID simply degenerates to Euclidean distance.

The complexity of a time series can be estimated in different ways, as discussed in Parmezan and Batista (2015). The original CID measure uses a fairly simple complexity estimation. It is based on the physical intuition that if we could "stretch" a time series until it becomes a straight line. Thus, a complex time series would result in a longer line than a simple time series. The complexity estimate can be computed using Eq. (10).

$$CE(T) = \sqrt{\sum_{i=1}^{n-1} (t_i - t_{i+1})^2} \tag{10}$$

Fig. 12 illustrates the intuition behind the idea of complexity estimated considered by CID with some examples.

5. Conformal Prediction

As previously discussed, there are empirical evidence which strongly suggest the simple nearest neighbor algorithm using an adequate distance measure outperforms more "sophisticated" time series classification methods in a wide range of application domains (Ding et al., 2008; Wang et al., 2013). For this reason, the 1NN with different distance measures was chosen for the classification task explored in this work.

The simple 1NN algorithm only returns the class label for an unknown example and their distance for the most similar example from the labeled training set. Thus, given their simplicity, we cannot measure how wrong the prediction is, it is simply right or wrong (Shafer and Vovk, 2008). The absence of this information can difficult their use in some approaches such as combining multiple classifiers.

As in this paper is proposed the asphalt pavement classification by combining time series classifiers built from the three axes of an accelerometer sensor, in this section is presented a precise way to determine the confidence level of the 1NN predictions, called *Conformal Prediction* (CP) (Shafer and Vovk, 2008; Vovk et al., 2005).

Given a specified probability of error ϵ, for example $\epsilon = 0.05$, CP output a prediction set that contains the actual label with a confidence level at least $(1-\epsilon)$ of probability for an unseen example. This prediction set is called *prediction region* Γ^{ϵ}.

Conformal Prediction uses past experience to determine the levels of confidence in new predictions. The main idea of CP is to test how well new examples fit to previously data (training set) or in other words, how "strange" the new example is in comparison with the training set. The only assumption made is the new example to be predicted and the examples of the training set have been generated independently from the same distribution. Thus, in contrast to traditional statistical methods based on parameterized models, no particular assumptions are made regarding the structure of the underlying probability distribution (Laxhammar and Falkman, 2010).

The method can be used for the learning tasks of classification and regression. In classification, the algorithm estimate p-values for each possible label $y \in Y$ for a new example and exclude from the prediction set those labels to have a p-value less than the specified significance level ϵ. The p-value is an analogy with the statistical hypothesis testing that rejects a hypothesis with a specified significance level (Laxhammar and Falkman, 2010). In order to estimate these p-values, a *nonconformity measure* is used.

The nonconformity measure is a real-valued function $\mathcal{A}(B,z)$ that measures how different an example z is from the examples from a set in B. A nonconformity measure can be extracted from any machine learning algorithm including Support Vector Machines, Decision Trees, Neural Networks, and Nearest Neighbors. There is no universal method to choose the nonconformity measure, but the prediction regions will be efficient only if $\mathcal{A}(B,z)$ measures well how different z is from the examples in B.

Given the advantages of nearest neighbor for time series classification, this algorithm was chosen for the nonconformity measure. The nonconformity based on the distances of nearest neighbor has been proposed for classification problems according to Eq. (11) (Gammerman and Vovk, 2007).

$$\mathcal{A}(B,z) = \frac{\sum_{j=1}^{k} d_{zj}^{y}}{\sum_{j=1}^{k} d_{zj}^{\neg y}} \tag{11}$$

This measure is calculated for each class label y of the problem, where $\sum_{j=1}^{k} d_{zj}^{y}$ is the sum of distances of the k nearest examples to the example z with the same class label, and $\sum_{j=1}^{k} d_{zj}^{\neg y}$ is the sum of distances of the k nearest examples to z from different classes.

The CP algorithm calculates the nonconformity measure for all examples (training set and test example) and stores in a set α. With this values, the algorithm calculates the ratio of examples in the training set that shows a nonconformity measure larger than the value achieved by the test example. This ratio is the p-value of the example given a class label. If this ratio is larger than the probability error ϵ defined by the user, the p-value is add in the prediction region Γ^{ϵ} with a confidence level of $(1 - \epsilon)$. The steps of the Conformal Prediction is shown in Algorithm 1.

6. Proposed approach

Given the central hypothesis that the acceleration data collected by smartphone sensors are sufficiently representative for the identification

Algorithm 1: Conformal Prediction

```
Input: Nonconformity measure $\mathcal{A}$, significance level $\epsilon$, training
       set $(x_1, y_1), \dots, (x_n, y_n)$, test example $x_{n+1}$.
Output: Prediction set $\Gamma^\epsilon$ for $y_{n+1}$.
$\Gamma^\epsilon = \emptyset$
foreach $y \in Y$ do
    $y_{n+1} = y$
    $D = \{(x_1, y_1), \dots, (x_n, y_n), (x_{n+1}, y_{n+1})\}$
    for $i = 1, \dots, n+1$ do
        $\alpha_i = \mathcal{A}(\{(x_j, y_j) \in D \mid j \neq i\}, (x_i, y_i))$
    end
    $p_y = \frac{\left|\{i=1,\dots,n+1 \mid \alpha_i \geq \alpha_{n+1}\}\right|}{n+1}$
    if $p_y > \epsilon$ then
        $\Gamma^\epsilon = \{\Gamma^\epsilon, y_{n+1}\}$
    end
end
```

of asphalt pavement, the use of time series classification techniques must be adequate for this problem.

As previously reported, the literature indicates that a simple One-Nearest Neighbor algorithm, with a proper distance function, frequently outperforming more complex classification algorithms. This only leave opens the question of which distance measure to use, that basically depends on the domain.

In order to find the most adequate distance measure for our problem, were initially evaluated four popular distance measures: (*i*) Dynamic Time Warping (DTW), (*ii*) Longest Common Subsequence Similarity (LCSS), (*iii*) Derivative Dynamic Time Warping ($\mathrm{DD_{DTW}}$), and (*iv*) Derivative Transform Distance ($\mathrm{DTD_{DTW}}$). The choice of these distance measures was due to the need of warping invariance, given to distortions present in our acceleration data. The distortion can occur due to the different velocities during data collection or even due to the properties of the pavement.

However, in this paper we also have the hypothesis that the warping invariance is not sufficient to accomplish the proposed classification task. From a visual analysis of our data, we note that the classification of pavement also needs a distance measure invariant to complexity. As in other application domains, different classes may have examples with different complexities, and pairs of complex examples, even those which may seem very similar, tend to be further apart under current distance measures than pairs of simple examples. This fact introduces errors in nearest neighbor classification, where complex examples are incorrectly assigned to a simpler class.

In this paper is proposed the use of a warping distance measure combined with a distance invariant to complexity to perform the classification of asphalt pavement over different scenarios. Thus, were evaluated the combination of DTW, LCSS, $\mathrm{DD_{DTW}}$, and $\mathrm{DTD_{DTW}}$ with the Complexity Invariant Distance (CID). CID proved to be very efficient in a comprehensive set of time series classification problems with improvements in the accuracy of many cases (Batista et al., 2014). Although CID allows their use with any distance measure, the works from literature never used distance measures as LCSS, $\mathrm{DD_{DTW}}$, and $\mathrm{DTD_{DTW}}$ with a complexity distance.

As the data is constituted by the acceleration force A in three perpendicular directions (x, y, z), resulting in the time series A_x, A_y, and A_z, a classifier for each acceleration axis can be learned. Thus, if we consider the four distance measures evaluated in this paper, combined or not with CID, and the three acceleration axis, we have a total of 24 different classifiers. On the other hand, if we convert the accelerations A_x, A_y, A_z in a unique time series A_m that represents the acceleration magnitude, the number of possible classifiers is reduced to 8 (one classifier for each distance measure).

At this moment, a question is raised: the use of classifiers learned for each acceleration axis can contribute to better classifications or the acceleration magnitude A_m is sufficient and shows equivalent results?

In order to answer this question, is proposed the combination of classifiers learned from the axes A_x, A_y, A_z, and the results of these classifiers are compared to the classifiers learned from the acceleration magnitude A_m. To combine the classifiers, four popular rules as considered (Kittler et al., 1998):

- Majority vote (voting): choose the most commonly predicted class label by different classifiers;
- Minimum distance (mindist): choose the class label of the classifier that reported the minimum distance among the nearest distances from all the classifiers;
- Sum: sum the score value given by different classifiers for each class label and choose the label that accumulated the higher score;
- Product (prod): multiplies the score value given by different classifiers for each class label and choose the label that accumulated the higher score.

As the simple 1NN classifier does not output a score for each class label given a test example, is proposed the use of Conformal Prediction to obtain a confidence value and the use of *sum* and *product* rules for combining the classifiers. To the best of our knowledge, the use of Conformal Prediction to combine classifiers has never been used in the literature for multi-dimensional time series classification.

In Fig. 13, is shown a diagram that summarizes the proposed approach since the data collection phase until the experimental evaluation.

7. Experimental evaluation

In the experimental evaluation, 4 distance measures were evaluated (DTW, LCSS, $\mathrm{DD_{DTW}}$, $\mathrm{DTD_{DTW}}$) alone and in conjunction with CID on the acceleration data from the axes x, y, z and from the acceleration magnitude.

For each distance measure, the results of the classifiers from the acceleration axes x, y, z were combined using five combining rules (*voting, mindist, sum, prod*, and *max*). In the case of the use of acceleration magnitude, it was generated a class label for each distance measure. In total, it was generated 48 results.

Two-fold cross-validation was used to partition the datasets into training and test sets. In order to reduce results variance by chance, this process was repeated five times, randomizing the order of examples between executions, i.e., was performed 5 × 2-fold cross-validation as suggested by Dietterich (1998).

The results of each scenario considered in this paper are presented in the next sections.

7.1. Results for the asphalt-regularity

In the first scenario, the goal of the classification task is the identification of regular or deteriorated pavement asphalt.

The average accuracy and standard error achieved on the five runs of two-fold cross-validation for the Asphalt-Regularity are shown in Table 5. In this table, the best result achieved by a distance measure given a combining rule is highlighted in bold. For example, for *voting* rule the best result is achieved by CID-LCSS measure with an average classification accuracy of 97.98% and 0.63 of standard error. The best result achieved by a combining rule given a distance measure is underlined. For example, for DTW distance measure the best result is achieved by the *mindist* combining rule, with an average classification accuracy of 69.83% and 1.20 of standard error.

We can note in Table 5 for the Asphalt-Regularity dataset, the best result was achieved by the distance measure LCSS combined with the Complexity Invariant Distance (CID-LCSS). In this case, both the *sum*

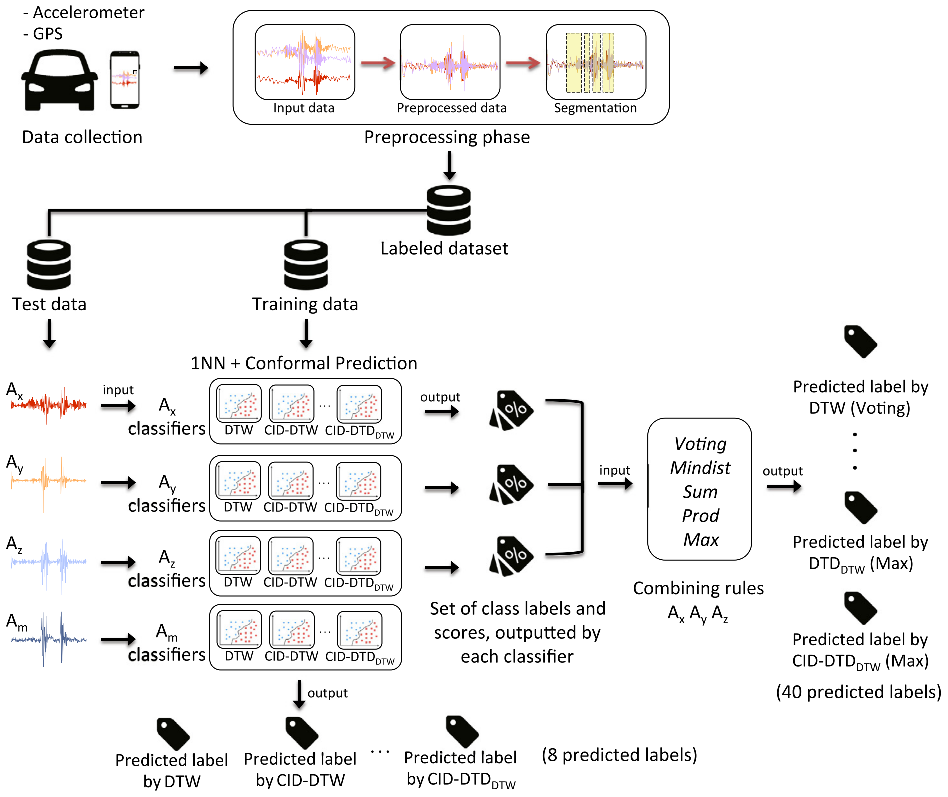


Fig. 13. Diagram of the proposed approach for the classification of asphalt pavement using smartphone sensors.

Table 5
Average accuracy (standard error) achieved by different distances measures and combining rules for the Asphalt-Regularity dataset.

Distance measure	Combining rule (A_x, A_y, A_z)					Accel. Mag. (A_m)
	Voting	Mindist	Sum	Prod	Max	
DTW	67.00 (1.13)	69.83 (1.20)	49.29 (0.11)	49.27 (0.09)	49.29 (0.06)	68.20 (2.10)
CID-DTW	97.23 (0.47)	95.63 (0.86)	93.00 (0.38)	92.92 (0.37)	92.44 (0.53)	92.46 (0.56)
LCSS	89.64 (0.69)	86.98 (1.30)	91.50 (0.67)	91.50 (0.67)	91.56 (0.49)	88.23 (1.04)
CID-LCSS	**97.98** (0.63)	95.29 (0.68)	**98.48** (0.45)	**98.48** (0.45)	**98.43** (0.50)	**96.48** (0.24)
DD_{DTW}	82.61 (2.68)	83.62 (2.37)	49.37 (0.26)	49.35 (0.20)	49.40 (0.37)	77.44 (2.85)
CID-DD_{DTW}	97.68 (0.20)	**96.86** (0.79)	94.81 (1.44)	94.75 (1.27)	94.22 (1.33)	95.42 (0.63)
DTD_{DTW}	73.87 (3.88)	77.31 (3.26)	49.35 (0.12)	49.35 (0.12)	49.35 (0.12)	68.20 (2.51)
CID-DTD_{DTW}	97.68 (0.28)	96.54 (0.54)	94.09 (1.22)	94.03 (1.20)	93.18 (1.29)	94.99 (1.50)

and *prod* combining rules of the acceleration data (A_x, A_y, A_z) shown 98.45% of accuracy. In the case of the use of acceleration magnitude A_m, CID-LCSS showed an equivalent result of 96.48%.

On the other hand, the worst result noted in Table 5 (49.27%), was achieved by the distance DTW with the *prod* combining rule. For this distance, the best strategy is to choose the class label of the classifier that reported the minimum distance among the acceleration data A_x, A_y, A_z, given by *mindist*.

From the analysis of the results in Table 5, it is also possible to note the contribution of the CID for all distance measures. For example, consider the result achieved by DTW with *voting* combining rule. DTW shows an average accuracy of 67%, while CID-DTW shows 97.23% of accuracy. This value represents a gain superior to 30%. Similar results are also obtained by other distance measures. This fact is a strong evidence of the need of a complexity invariant distance to classify this kind of data.

In Table 6, is presented the confusion matrix (or error matrix) of the best classifier for the Asphalt-Regularity dataset, which is CID-LCSS with *sum* combining rule. Each column of the matrix represents the amount of instances in a predicted class while each row represents the amount of instances in an actual class. All correct predictions located in the diagonal of the matrix was highlighted in bold. In Table 6, the evaluation measures Precision, Recall and F-Measure (F1) for each class, are also shown. The results shown in this matrix are from one of the five runs of the cross-validation performed in the experimental evaluation.

Table 6
Confusion matrix and detailed results achieved by the distance measure CID-LCSS and *sum* combining rule for the Asphalt-Regularity dataset.

		Predicted		Precision	Recall	F1
		Reg	Det			
Actual	Regular (Reg)	**373**	8	0.99	0.98	0.99
	Deteriorated (Det)	3	**367**	0.98	0.99	0.99

7.2. Results for the Asphalt-PavType

In the second scenario, the goal of the classification task is the identification of pavement type between three different classes (flexible, cobblestone, and dirt road), considered in the Asphalt-PavType dataset.

The results achieved by different distance measures and combining rules for the Asphalt-PavType dataset are shown in Table 7.

Table 7
Average accuracy (standard error) achieved by different distances measures and combining rules for the Asphalt-PavType dataset.

Distance measure	Combining rule (A_x, A_y, A_z)					Accel. Mag. (A_m)
	Voting	Mindist	Sum	Prod	Max	
DTW	61.55 (1.14)	64.42 (1.06)	41.08 (3.91)	40.19 (3.56)	30.43 (3.75)	58.18 (2.05)
CID-DTW	82.41 (0.74)	78.86 (0.52)	85.16 (0.62)	85.31 (0.63)	82.14 (0.93)	77.23 (1.73)
LCSS	69.00 (1.29)	67.17 (1.11)	73.14 (1.29)	73.12 (1.21)	72.15 (1.16)	67.11 (1.39)
CID-LCSS	**83.66** (0.87)	**79.55** (0.70)	**88.17** (1.33)	**88.27** (1.33)	**86.54** (0.80)	**80.66** (1.25)
DD_{DTW}	63.28 (1.58)	67.91 (1.59)	41.14 (4.23)	40.61 (3.90)	31.83 (4.43)	59.56 (1.95)
CID-DD_{DTW}	83.36 (1.04)	78.43 (0.53)	86.84 (0.97)	87.11 (0.95)	84.02 (0.99)	79.13 (3.15)
DTD_{DTW}	67.07 (1.97)	70.09 (1.29)	44.63 (4.27)	44.09 (3.97)	29.08 (1.30)	61.61 (2.52)
CID-DTD_{DTW}	82.84 (0.82)	78.43 (1.64)	86.73 (0.35)	86.92 (0.65)	83.47 (1.15)	79.01 (1.53)

Table 8
Confusion matrix and detailed result achieved by the distance measure CID-LCSS and Prod combining rule for the Asphalt-PavType dataset.

		Predicted			Precision	Recall	F1
		Flex	Cobb	Dirt			
Actual	Flexible (Flex)	**401**	0	7	0.99	0.98	0.99
	Cobblestone (Cobb)	0	**222**	41	0.74	0.84	0.79
	Dirt road (Dirt)	4	78	**302**	0.86	0.79	0.82

In Table 7, we can note as well as in Asphalt-Regularity dataset, the distance CID-LCSS with the *prod* combining rule, showed the best accuracy result for the Asphalt-PavType dataset (88.27%). Given the use of the acceleration magnitude data, the best result was also achieved by CID-LCSS, with 80.66%. For this dataset, the worst result was achieved by the distance DTD_{DTW} with the *max* combining rule (29.08%). It is also noted a substantial contribution of CID for all other measures.

The confusion matrix of the 1NN classifier with CID-LCSS as distance measure and the use of *prod* rule to combine the outputs of classifiers generated from the acceleration data A_x, A_y, A_z, are shown in Table 8.

From the analysis of the prediction errors showed in Table 8, it is noted that one of the most difficult of the classifier is the correct identification of the cobblestone class, where 41 examples of a total of 263 are incorrectly classified as a dirt road. Even so, the classification accuracy for the cobblestone class was 84.41%.

7.3. Results for the Asphalt-Obstacles

In the last scenario evaluated in this paper, the goal of the classification task in the Asphalt-Obstacles dataset is the identification of obstacles in the road, as speed bump, vertical patch, raised markers, and raised crosswalk.

The results for the Asphalt-Obstacles dataset are shown in Table 9.

The best result for the Asphalt-Obstacles dataset was achieved by DTW distance using the acceleration magnitude data, with 81.13% of accuracy. Although the best result was achieved by the classifier with the acceleration magnitude data, if we consider the standard error, the distance CID-LCSS showed an equivalent result to the best classifier (DTW) with 79.64% (2.03), considering the *sum* combining rule. On the other hand, the worst result was achieved by the distance measure DTD_{DTW} with *max* combining rule, where was achieved 43.18% of accuracy.

It is interesting to note in Table 9 that unlike the other two datasets, the complexity invariant distance contributes only to the LCSS measure to achieve better results. In fact, from a visual inspection of the data from the three datasets, it is possible to observe a smaller variability of complexity between the examples of the same class for the Asphalt-Obstacles dataset. Thus, there is less possibility of complex examples to be incorrectly assigned to a simpler class.

This fact is better illustrated in Fig. 14, where are presented the complexities (according to Eq. (10)) of the class examples from the Asphalt-PavType and Asphalt-Obstacles datasets. In order to better visualize, is shown only the function that fit the histogram of each class.

In Fig. 14-(b) we can note the complexity of 3/4 of the classes of the Asphalt-Obstacles dataset slightly varies from 0 to 30 and a multimodal distribution for the class *raised markers*. On the other hand, for the Asphalt-PavType dataset, the complexity of the examples of 2/3 of the classes are substantially higher and varies from 10 to 80. In fact, the confusion matrix previously presented in Table 8, shows the errors concentrated in these more complex classes.

The confusion matrix with the predictions of the best classifier for the Asphalt-Obstacles dataset is shown in Table 10. In this table, it is noted most of the errors concentrated in the class *vertical patch* and *raised markers*. This error was expected, given that some raised markers with a reduced size are responsible for generating an impact in the vehicle that is similar to those generated by pavement patches.

7.4. Analysis and discussion

From the results previously shown for each dataset, some questions can be raised for the classification problem of asphalt pavement using accelerometer data. For example:

- Among different distance measures evaluated, which ones are better to discriminate our data?
- The complexity invariance really improve the results of other distance measures for our data?
- We need combine different classifiers generated from each acceleration axes or the acceleration magnitude is sufficient to achieve similar results?

In order to answer these questions, the Friedman test with the Nemenyi post hoc with 95% as confidence level was performed considering all results. This test allows to statistically compare the results achieved by different distance measures and combining rules.

The critical difference diagram that illustrates the results of the statistical test considering the evaluation of different distance measures, is shown in Fig. 15. In this diagram, the methods are sorted according to their average ranking and the methods connected by a line do not present statistically significant differences among them.

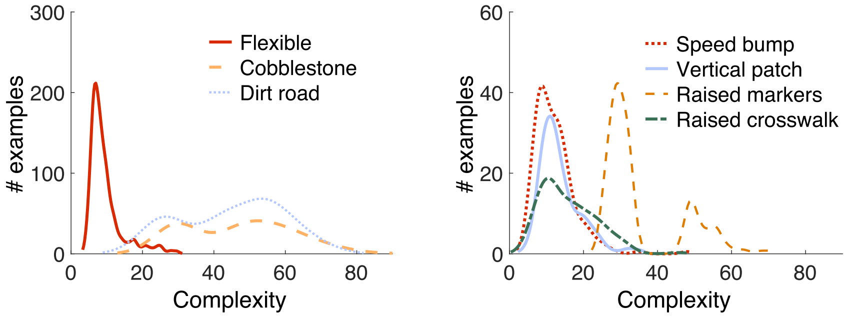
From the analysis of diagram shown in Fig. 15, we can answer the first two questions raised at the beginning of this section. First, there is no statistical difference between CID-LCSS and CID-DD_{DTW}, but CID-LCSS was ranked in the first position. In addition, both methods showed superior results than the other measures with statistical difference. Second, all distance measures statistically improve their results with CID.

Table 9
Average accuracy (standard error) achieved by different distances measures and combining rules for the Asphalt-Obstacles dataset.

Distance measure	Combining rule (A_x, A_y, A_z)					Accel. Mag. (A_m)
	Voting	Mindist	Sum	Prod	Max	
DTW	64.97 (2.93)	54.20 (1.51)	47.59 (2.74)	47.23 (3.00)	47.69 (3.21)	**81.13** (1.29)
CID-DTW	57.85 (0.53)	53.90 (1.96)	61.28 (2.84)	61.49 (2.75)	59.13 (1.04)	74.21 (1.18)
LCSS	66.56 (1.77)	60.82 (2.06)	77.64 (1.41)	77.44 (1.07)	76.26 (2.54)	77.90 (1.08)
CID-LCSS	**74.56** (1.42)	**68.72** (1.06)	**79.64** (2.03)	**79.44** (2.03)	**77.85** (2.12)	75.95 (2.61)
DD_{DTW}	63.79 (0.98)	56.05 (2.26)	47.49 (2.08)	47.54 (2.00)	46.61 (1.64)	75.85 (2.27)
CID-DD_{DTW}	57.44 (2.65)	52.97 (1.36)	60.21 (2.55)	59.79 (2.87)	58.05 (2.74)	68.56 (4.08)
DTD_{DTW}	65.28 (2.05)	56.21 (2.53)	44.15 (1.75)	44.26 (1.87)	43.18 (1.60)	75.59 (2.82)
CID-DTD_{DTW}	55.54 (0.82)	51.85 (1.98)	56.10 (1.14)	55.74 (0.82)	55.64 (1.58)	66.67 (2.83)

Table 10
Confusion matrix and detailed result achieved by the classifier that uses only the acceleration magnitude (A_m) with Dynamic Time Warping for the Asphalt-Obstacles dataset.

		Predicted				Precision	Recall	F1
		SB	VP	RM	RC			
Actual	Speed bump (SB)	**99**	3	1	3	0.87	0.93	0.90
	Vertical patch (VP)	6	**75**	23	7	0.80	0.68	0.73
	Raised markers (RM)	1	14	**75**	3	0.76	0.81	0.78
	Raised crosswalk (RC)	8	2	0	**70**	0.84	0.88	0.86



(a) Asphalt-PavType (b) Asphalt-Obstacles

Fig. 14. Class complexity estimates of Asphalt-PavTyp and Asphalt-Obstacles datasets.

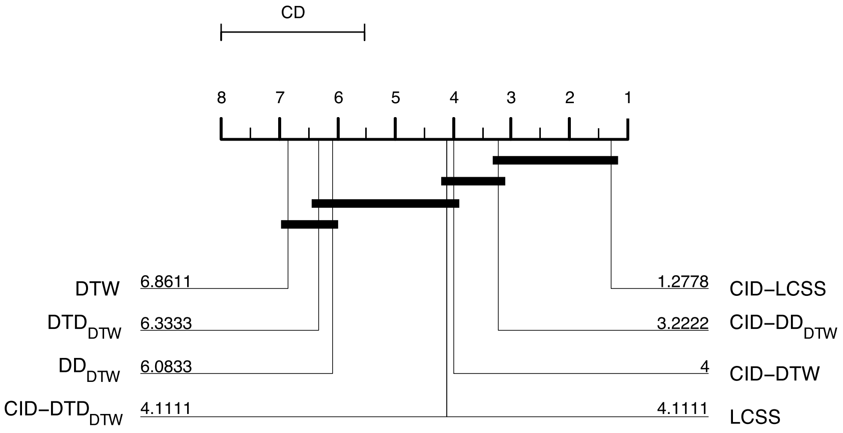


Fig. 15. Critical difference diagram considering the average results achieved by the 1NN classifier with different distances measures.

As pointed by Vlachos et al. (2003), the flexibility provided by DTW is very important, however its efficiency deteriorates for noisy data since by matching all the points, it also matches the outliers distorting the true distance between the sequences. Given the characteristics of our data and presence of noise, LCSS is an adequate distance measure. LCSS provide the match of two sequences by allowing them to stretch, without rearranging the order of the observations but allowing some elements to be unmatched. Thus, some of noise observations are ignored by the distance.

To answer the last question, we analyze the diagram shown in Fig. 16, where we compare the results of different combining rules and the results achieved by the classifiers using the acceleration magnitude.

We noted in Fig. 16 that the *voting* rule showed the best result among the evaluated combining rules. However, this result is not statistically

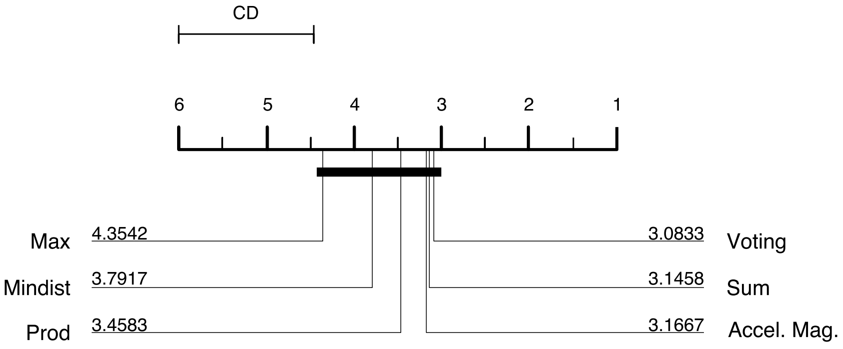


Fig. 16. Critical difference diagram considering the average results achieved by different combining rules and the classifier which considers only the acceleration magnitude.

superior to the other combining rules or the use of a classifier which considers only the acceleration magnitude. Thus, the choice of a combining rule is a less important decision to achieve better results in our problem.

In summary, among some possible configurations to perform the classification task of asphalt pavement using data gathered by the accelerometer, we can suggest the use of a 1NN classifier with the distance measure CID-LCSS or CID-DD$_{\mathrm{DTW}}$. The main advantage of CID-LCSS is that CID considers an implicit feature of the data related to the complexity of the series which is very representative of the problem. In addition, LCSS allows the non-linear alignment of the series, ignoring noise observations.

The classifier can use only the information of acceleration magnitude or the acceleration of the axes x, y, z. In the last case, we suggest the use of *sum* or *voting* rule to combine the confidences outputted by the classifiers from each axis. However, the choice of a combining rule is a less important decision. The advantages of the use of acceleration magnitude are related to the lower volume of data generated and stored on the smartphone and to the lower time for processing the data, given that three time series from the different axes are converted into a unique series.

8. Conclusions and future works

The large variety of sensors in most modern smartphones makes these devices useful tools to collect a large amount of data in many situations, allowing the emergence of new systems and applications. In this paper, a real data stream problem of asphalt evaluation using acceleration data gathered by smartphones was presented.

The problem of asphalt evaluation was modeled in this work as a multi-dimensional time series classification task, where the time series are the data of the three axes accelerometer sensor of the smartphone.

We provide a comprehensive experimental evaluation of different distance measures for time series, as DTW, LCSS, DD$_{\mathrm{DTW}}$, DTD$_{\mathrm{DTW}}$, alone and in conjunction with CID. In addition, different combining rules also were evaluated in order to use data from different acceleration axis. All these methods were evaluated on three datasets collected in this work and which represents different classification problems.

One hypothesis of this paper is that the warping invariance, provided by distance measures as DTW, is not sufficient to accomplish the proposed classification task. Thus, we experimentally show that the complexity invariance improves the results achieved by classical distance measures. Specifically, we suggest the use of the distance measure LCSS in conjunction to CID. To combine classifiers from each axis, we suggest the use of *sum* or *voting* rules.

In future works, we intend to collect more data considering different vehicles to evaluate the effectiveness of the classifiers on diversified conditions. We also want to include new classes to the datasets, as different levels of pavement quality as perfect, good, average, poor, and fair. A change of representation frequently reveals features that are not apparent in the original data representation. Thus, in future works we will explore different data representations, as those provided by Discrete Fourier Transform (DFT) and Discrete Wavelet Transform (DWT).

Acknowledgment

The author would like to thank the startup Onion Tecnologia where this work was conducted and São Paulo Research Foundation (FAPESP) for the financial support in the grant number 2016/07767-3.

References

Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E., 2016. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min. Knowl. Discov.

Batista, G.E.A.P.A., Keogh, E., Tataw, O.M., Souza, V.M.A., 2014. Cid: an efficient complexity-invariant distance for time series. Data Min. Knowl. Discov. 28 (3), 634–669.

Brisimi, T.S., Cassandras, C.G., Osgood, C., Paschalidis, I.C.H., Zhang, Y., 2016. Sensing and classifying roadway obstacles in smart cities: The street bump system. IEEE Access 4, 1301–1312.

Dietterich, T.G., 1998. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput. 10 (7), 1895–1923.

Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E., 2008. Querying and mining of time series data: experimental comparison of representations and distance measures. Proc. VLDB Endowment 1 (2), 1542–1552.

Engelbrecht, J., Booysen, M.J., van Rooyen, G.-J., Bruwer, F.J., 2015. Survey of smartphone-based sensing in vehicles for intelligent transportation system applications. IET Intel. Transport Syst. 9 (10), 924–935.

Eriksson, J., Girod, L., Hull, B., Newton, R., Madden, S., Balakrishnan, H., 2008. The pothole patrol: using a mobile sensor network for road surface monitoring. In: International Conference on Mobile Systems, Applications, and Services. pp. 29–39.

Gammerman, A., Vovk, V., 2007. Hedging predictions in machine learning the second computer journal lecture. Comput. J. 50 (2), 151–163.

Górecki, T., Łuczak, M., 2013. Using derivatives in time series classification. Data Min. Knowl. Discov. 26, 310–331.

Górecki, T., Łuczak, M., 2014. Non-isometric transforms in time series classification using dtw. Knowl.-Based Syst. 61, 98–108.

Hirschberg, D.S., 1975. A linear space algorithm for computing maximal common subsequences. Commun. ACM 18 (6), 341–343.

Jeong, Y.-S., Jeong, M.K., Omitaomu, O.A., 2011. Weighted dynamic time warping for time series classification. Pattern Recognit. 44 (9), 2231–2240.

Keogh, E.J., Pazzani, M.J., 2001. Derivative dynamic time warping. In: SIAM International Conference on Data Mining. pp. 1–11.

Kim, D.-W., Lee, J., Lim, H., Seo, J., Kang, B.-Y., 2014. Efficient dynamic time warping for 3d handwriting recognition using gyroscope equipped smartphones. Expert Syst. Appl. 41 (11), 5180–5189.

Kittler, J., Hatef, M., Duin, R.P., Matas, J., 1998. On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell. 20 (3), 226–239.

Latecki, L.J., Megalooikonomou, V., Wang, Q., Lakaemper, R., Ratanamahatana, C.A., Keogh, E., 2005. Elastic partial matching of time series. In: European Conference on Principles of Data Mining and Knowledge Discovery. pp. 577–584.

Laxhammar, R., Falkman, G., 2010. Conformal prediction for distribution-independent anomaly detection in streaming vessel data. In: First International Workshop on Novel Data Stream Pattern Mining Techniques. pp. 47–55.

Liu, J., Zhong, L., Wickramasuriya, J., Vasudevan, V., 2009. uwave: Accelerometer-based personalized gesture recognition and its applications. Pervasive Mob. Comput. 5 (6), 657–675.

Mednis, A., Strazdins, G., Zviedris, R., Kanonirs, G., Selavo, L., 2011. Real time pothole detection using android smartphones with accelerometers. In: International Conference on Distributed Computing in Sensor Systems and Workshops. pp. 1–6.

Mohan, P., Padmanabhan, V.N., Ramjee, R., 2008. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In: ACM Conference on Embedded Network Sensor Systems. pp. 323–336.

Mohod, M.V., Kadam, K., 2016. A comparative study on rigid and flexible pavement: A review. IOSR J. Mech. Civ. Eng. 13 (3), 84–88.

Parmezan, A.R.S., Batista, G.E.A.P.A., 2015. A study of the use of complexity measures in the similarity search process adopted by knn algorithm for time series prediction. In: IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 45–51.

Predic, B., Stojanovic, D., 2015. Enhancing driver situational awareness through crowd intelligence. Expert Syst. Appl. 42 (11), 4892–4909.

Ronao, C.A., Cho, S.-B., 2016. Human activity recognition with smartphone sensors using deep learning neural networks. Expert Syst. Appl. 59, 235–244.

Sakoe, H., Chiba, S., 1978. Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoust. Speech Signal Process. 26 (1), 43–49.

Selesnick, I.W., Burrus, C.S., 1998. Generalized digital butterworth filter design. IEEE Trans. Signal Process. 46 (6), 1688–1694.

Shafer, G., Vovk, V., 2008. A tutorial on conformal prediction. J. Mach. Learn. Res. 9 (Mar), 371–421.

Silva, D.F., Souza, V.M.A., Batista, G.E.A.P.A., 2013. Time series classification using compression distance of recurrence plots. In: IEEE International Conference on Data Mining (ICDM). pp. 687–696.

Souza, V.M.A., Cherman, E.A., Rossi, R.G., Souza, R.A., 2017. Towards automatic evaluation of asphalt irregularity using smartphones sensors. In: International Symposium on Intelligent Data Analysis. pp. 322–333.

Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., Keogh, E., 2003. Indexing multi-dimensional time-series with support for multiple distance measures. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 216–225.

Vovk, V., Gammerman, A., Shafer, G., 2005. Algorithmic Learning in a Random World. Springer Science & Business Media.

Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P., Keogh, E., 2013. Experimental comparison of representation methods and distance measures for time series data. Data Min. Knowl. Discov. 1–35.