sensors

MDPI

Article

Environment Monitoring for Anomaly Detection System Using Smartphones †

Van Khang Nguyen [1,2,*], Éric Renault [1] and Ruben Milocco [3]

1 SAMOVAR, CNRS, Télécom SudParis, Institut Polytechnique de Paris, 91011 Évry, France
2 College of Education, Hue University, 530000 Hue, Vietnam
3 GCAyS, UNComahue, Buenos Aires 1400, 8300 Neuquén, Argentina
* Correspondence: nguyenvankhang@yahoo.com

† This paper is an extended version of an earlier conference paper: Van Khang, N.; Renault, É. Cooperative Sensing and Analysis for a Smart Pothole Detection. In Proceedings of the 15th International Wireless Communications Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019.

Received: 2 August 2019; Accepted: 19 August 2019; Published: 5 September 2019

Abstract: Currently, the popularity of smartphones with networking capabilities equipped with various sensors and the low cost of the Internet have opened up great opportunities for the use of smartphones for sensing systems. One of the most popular applications is the monitoring and the detection of anomalies in the environment. In this article, we propose to enhance classic road anomaly detection methods using the Grubbs test on a sliding window to make it adaptive to the local characteristics of the road. This allows more precision in the detection of potholes and also building algorithms that consume less resources on smartphones and adapt better to real conditions by applying statistical outlier tests on current threshold-based anomaly detection methods. We also include a clustering algorithm and a mean shift-based algorithm to aggregate reported anomalies on data to the server. Experiments and simulations allow us to confirm the effectiveness of the proposed methods.

Keywords: anomaly detection; anomalies aggregation; smartphone sensing; sensor networks

1. Introduction

The mobile phone sensor system is promising great potential for applications. For the past few years, smartphones have become more popular and powerful. Any smartphone today contains many different sensors that sense information from the surrounding environment such as a camera, a microphone, a GPS sensor, an accelerometer, a proximity sensor, an ambient light sensor, a magnetometer, a barometer, an air humidity sensor, a thermometer, etc. Besides, network costs are getting lower and lower. Therefore, the research and application of the mobile phone sensor system has become increasingly popular.

Many studies of smartphone sensors and mobile phone sensor system have been described in the literature [1–3]. Common areas of application are: personal health monitoring, calculation of environmental impact, fall detection, freezing of gait detection, monitoring and traffic conditions, monitoring noise and ambiance, etc.

It is possible to classify areas such as monitoring road and traffic conditions, monitoring noise, and ambiance and fall detection into monitoring and anomaly detection systems. In the future, there will be many other applications that can be exploited from monitoring and anomaly detection systems based on data from sensors such as ambient light sensors, magnetometers, barometers, air humidity sensors, and thermometers. The basic processing of monitoring and anomaly detection systems is as follows: Data from sensors are collected regularly in time. These data on smartphones are comprised of

anomalies such as road anomalies, falls, or unusual sounds. If it is necessary to compute environmental conditions, such as road flatness or noise, depending on the type of applications, all data or only the remaining part of data is analyzed. All final required results are sent to the server for further processing. This paper presents the overall architecture of the system we developed for the monitoring and the detection of anomalies. With this model, we propose lightweight algorithms to identify anomalies on smartphones and an algorithm to aggregate these anomalies and the associated data on the server. We have also conducted experiments to validate the approach and evaluate the results of the proposed algorithms. To evaluate our road anomaly detection algorithms, we applied them during experiments on real data. For the specific case of the anomaly aggregation algorithms used to decide whether anomalies are true potholes or artifacts, we performed simulations to evaluate the results because it was not possible to get enough real data for each pothole.

The rest of the paper is organized as follows: Section 2 introduces our smartphone-server architecture for environment anomaly detection. Section 3 presents the way to apply the Grubbs test on threshold-based anomaly detection algorithms and experiments to verify this method. Section 4 describes the problem of anomaly data aggregation, our solution, and the simulation method. Finally, Section 5 concludes the paper.

2. System Architecture

In our anomaly detection system, we divided the anomaly detection task into several stages. Some of these stages are handled at the server to reduce the power consumption of smartphones (see Figure 1). The functionality of the system components are described as follows.

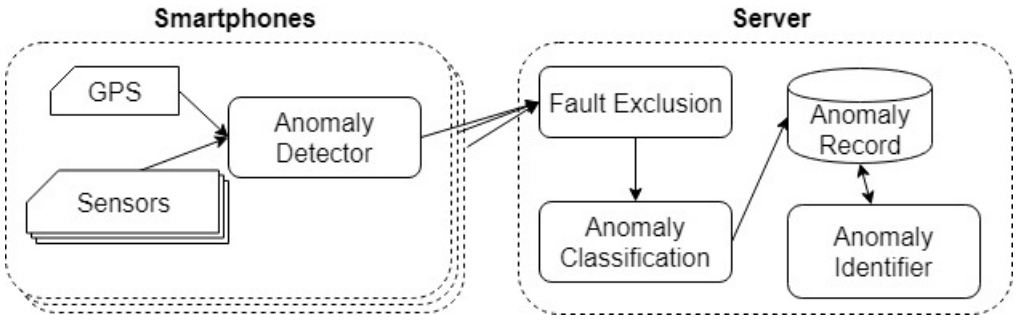

Figure 1. Smartphone-server system architecture for road anomaly detection, © 2019 IEEE [4].

- The Anomaly Detector aims at initially detecting anomalies with a lightweight algorithm, i.e., an algorithm that consumes very few device resources. A resident program is installed on the phone to read the data from the accelerometer and the GPS sensor. These data are passed to the anomaly detection component under a certain condition, for example when the smartphone reaches a speed greater than a certain value for the case of the monitoring and traffic conditions. Then, the program receives a list of anomalies to send to the server when connected.
- The Fault Exclusion component is intended to eliminate false anomalies caused by user's actions. This component can be viewed as a detached part of the Anomaly Detector. The separation between the two components (Anomaly Detector and Fault Exclusion) is intended to distribute processing between the client and the server to optimize resource consumption of the smartphones.
- The Anomaly Classification component classifies anomalies, which also allows for the elimination of less reliable anomalies.
- The Anomaly Identifier component aggregates anomaly reports from multiple smartphones to locate anomalies and compute a confidence weight associated with each location.

In this paper, we focus on two important components: the Anomaly Detector and Anomaly Identifier.

3. Anomaly Detection Algorithm

Many different types of anomaly detection methods have been proposed, depending on the purpose and type of sensor data used. In this section, we propose an improvement for road anomaly detection methods based on acceleration data, also known as vibration-based methods.

3.1. Related Works

Current vibration-based methods for road anomaly detection using smartphones can be divided into two classes: threshold-based methods and classification-based methods [5]. Threshold-based methods are simple algorithms to identify potholes by verifying if a component or a value derived from a function of a measured acceleration exceeds a specific threshold. With the pothole patrol system, Jakob Eriksson et al. [6] proposed an algorithm based on a set of threshold-based filters. Prashanth Mohan et al. [7] proposed the the Nericell road monitoring system in which they detected braking by comparing the average value of the x component of the acceleration (the projection on the axis in the direction of the motion) in a sliding window to a threshold. They detected bumps and potholes by comparing the z component of the acceleration (the projection on the vertical axis) to two threshold values depending on the speed. Astarita Vittorio et al. [8] proposed to compare the distance between two extreme acceleration values (min and max) with a threshold to detect road anomalies. Artis Mednis et al. [9] proposed four threshold-based algorithms Z-THRESH , Z-DIFF , STDEV(Z), G-ZERO (see below). In classification-based methods, several features such as the mean, variance, and root mean square of the three axes, as well as the correlation between the axes [10–12] are first extracted from the acceleration data. Second, a machine-learning algorithm typically using Support Vector Machines (SVM) is applied to classify these features into road anomaly or artifact. We found that the current methods were less adaptable to real conditions such as the flatness of the road, the type of vehicle, the speed of travel, the quality of the suspension, and smartphone type. For the threshold-based approach, the authors used a fixed threshold. Meanwhile, the fluctuation of acceleration data depended very much on real conditions. For classification-based methods, training was performed on a road segment with a vehicle with some characteristics. It was difficult to produce good results in this study when applied to vehicles with other characteristics.

We investigated how to improve threshold-based algorithms to achieve small complexity algorithms and the least phone resource consumption. We chose threshold-based algorithms because they require low resources, while classification-based algorithms require many resources, leading to much energy consumption. Obviously, the smartphone battery power must be reserved for user functions.

Here are the algorithms that we improved and on which we did experiments to validate their effectiveness:

- Four algorithms proposed by Artis Mednis et al. [9]:
  - **Z-THRESH**: If the amplitude of the value on the z-t axis of acceleration data is greater than a specified threshold, a road anomaly is detected.
  - With **Z-DIFF**, events are detected when the difference between two consecutive values is greater than a specific threshold.
  - **STDEV(Z)** is based on the standard deviation in a sliding window. When the standard deviation is greater than a specific threshold, an event is detected.
  - With **G-ZERO**, an event is detected when the values of all three axes are less than a specific threshold.
- The algorithm proposed by Vittorio et al. [8] uses the vertical acceleration provided by both the accelerometer and GPS sensor only. For simplicity, we refer to this algorithm as DVA-THRESH in the rest of this paper. Since the GPS data frequency was 1 Hz and one of the accelerometers was at least 5 Hz, the authors preprocessed the accelerometer data in groups of one second

by computing a_{z_min}, a_{z_max}, and a_{z_avg}. The detection was based on the difference in vertical acceleration impulse defined by $DVA = a_{z\_max} - a_{z\_min}$.

The road anomaly filter is given by the following operation:

$$DVA = \begin{cases} 0 & \text{if } DVA \leq DVA^{\text{th}} \\ DVA & \text{if } DVA > DVA^{\text{th}} \end{cases}$$

where DVA^{th} is a reference set of DVA values that was determined by previous experiments.

3.2. Improvement of Anomaly Detection Algorithms

We considered the road anomaly through the abnormal sensor data we obtained and compared them to surrounding data. When the car enters a road anomaly like a pothole, the statistic moments of acceleration change. This is the basis for anomaly detection. However, the change of acceleration data depends on many factors. If a fixed threshold value is used, it is not possible to determine the abnormality correctly. "Abnormal data" here must be understood as abnormal when compared to the data before and after the car enters the pothole.

On this basis, we propose an anomaly detection on the small sample of sensor data lastly obtained by using a statistical outlier test method to find anomalies.

We chose to apply the Grubbs test method. This is a very popular outlier test method for univariate datasets. In particular, the application of this method allows us to reduce the computational complexity.

Application of the Grubbs' Test to Threshold-Based Anomaly Detection Algorithms

The Grubbs' test [13] is a statistical test used to detect outliers in a univariate dataset. The test assumes that the underlying data distribution is normal. Grubbs' test is defined when the following hypotheses are true:

- H_0: There is no outlier in the dataset.
- H_1: There is exactly one outlier in the dataset

The Grubbs' test statistic is defined as follows:

$$C = \frac{max_i |x_i - \overline{x}|}{\sigma}$$

where $\overline{x}$ and σ denote the sample mean and the standard deviation, respectively. For the two-sided test, the hypothesis of no outlier was rejected at the significance level α if:

$$C > G_{N,\alpha} = \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\alpha/(2N),N-2}}{N-2+t^2_{\alpha/(2N),N-2}}}$$

where $t^2_{\alpha/(2N),N-2}$ denotes the upper critical value of the t-distribution with $N-2$ degrees of freedom and a significant level of $\alpha/(2N)$. For the one-sided test, $\alpha/(2N)$ is replaced by α/N [14].

In order to apply the Grubbs test to the five algorithms mentioned above (namely Z-THRESH, Z-DIFF, STDEV(Z), G-ZERO, and DVA-THRESH), we compared the components of the acceleration data according to the Grubbs formula instead of comparing it with a fixed threshold. Let the data obtained from the acceleration sensor be a sequence denoted as $< t^i, a^i_x, a^i_y, a^i_z >$ where t^i is the time of sensing and a^i_x, a^i_y, a^i_z are the three components of the acceleration vector at time t^i. The value to compare depends on the specific algorithm, and the last N comparison values are retained as sample X (see Table 1).

Table 1. Comparison values of the algorithms, © 2019 IEEE [4].

Algorithm	**Value** X_i	**Anomaly Condition**
Z-THRESH	a_z^{last}	$X_i > \overline{X} + \sigma * G_{N,\alpha}$ *or* $X_i < \overline{X} - \sigma * G_{N,\alpha}$
Z-DIFF	$\lvert a_z^{last} - a_z^{last-1} \rvert$	$X_i > \overline{X} + \sigma * G_{N,\alpha}$
STDDEV(Z)	$stddev(a_z^{i-winSize} ... a_z^{last})$	$X_i > \overline{X} + \sigma * G_{N,\alpha}$
G -ZERO	$max(a_x^{last}, a_y^{last}, a_z^{last})$	$X_i < \overline{X} - \sigma * G_{N,\alpha}$
DVA-THRESH	$(a_{z\_max} - a_{z\_min})^{1sec}$	$X_i > \overline{X} + \sigma * G_{N,\alpha}$

We propose to process sensor data as a continuous sequence. Whenever a new acceleration data value is received, sample X is updated by adding the new value and removing the oldest one. At the same time, the mean value $\overline{X}$ and standard deviation σ_X must also be recomputed.

To reduce the complexity of the algorithm, we implemented the update of $\overline{X}$ and σ_X according to the cumulative method. Suppose X is updated by adding X_{new} (the new value in the sliding window) and deleting X_{old} (the oldest value in the sliding window), as $\sigma = \sqrt{\overline{X_i^2} - \overline{X}^2}$ (where $\bar{X}$ is the average of the value in X and $\bar{X}^2$ is the average of the squares of the values in X); one just needs to update $\overline{X}$ and $\overline{X^2}$ as follows:

$$\overline{X}_{new} = \overline{X}_{old} + \frac{X_{new} - X_{old}}{N}$$

$$\overline{X^2}_{new} = \overline{X^2}_{old} + \frac{X_{new}^2 - X_{old}^2}{N}$$

3.3. Experiment

3.3.1. Collection and Adjustment of Data

We collected accelerometer and GPS data on nearly 40 km of road in Hue city using both a Mazda 3 car and a Honda Lead 125 scooter. The smartphone used was a Wiko WIM Lite equipped with a 15.38-Hz accelerometer. We developed a program for the phone that allowed saving sensor data as a sequence of tuples like:

<time, 3-axis acceleration>

<time, location, speed>

To build the ground truth, we added a data collection application of buttons to allow marking every time the vehicle entered a road anomaly. There were 217 anomalies marked with the car and 219 anomalies marked with the scooter. We also built a software program to adjust the ground truth on the computers after data collection. The road anomalies were marked according to the timeline. Marking operations may be slower than the actual time the vehicle enters the anomaly. Therefore, we needed to adjust them manually to ensure accuracy.

3.3.2. Experiment Process and Results

We installed the five algorithms Z-THRESH, Z-DIFF, STDEV(Z), G-ZERO, and DVA-THRESH on the smartphone to test them. Each algorithm had two versions installed: the original version and the improved version with the Grubbs test method. Acceleration data were processed as a continuous data sequence, to simulate a real-time anomaly detection. The size of sample N in our experiment was 100. When an anomaly was detected, the corresponding data segment was compared to the ground truth data based on time. If there was an anomaly there, this was a True Positive (TP); otherwise, it was a False Positive (FP). A non-detected anomaly in the ground truths was a False Negative (FN).

In this experiment, the number of detected negative conditions, also known as True Negative (TN), cannot be determined. Note that data were processed as a continuous sequence; they were not cut into windows. Therefore, the number of negative cases was uncountable.

We chose the evaluation method based on the Precision (P) and Recall (R) parameters as defined by:

$$P = \frac{TP}{TP + FP} \text{ and } R = \frac{TP}{TP + FN}$$

Unlike methods such as the Receiver Operating Characteristic (ROC) curve [15], methods based on precision and recall are computed without the need for true negative values [16]. The experimental results were evaluated through the precision-recall curve and the F-measure index (also known as the F_1 score), which is the harmonic mean of both precision and recall:

$$F_1 = 2 \times \frac{P \times R}{P + R}$$

The analysis of the results on the precision-recall curves showed that the algorithms were providing better results when improved with the Grubbs test method. Figure 2 shows the corresponding precision-recall curves when applying the above five algorithms, both the original version and the improved version, to all data collected. Ideally, if there is no error, P and R should be equal to 1, i.e., the closer to the upper-right corner, the better the result. As shown in Figure 2, the graphs of the improved algorithms were all above, closer to the upper-right corner than the the original algorithms.

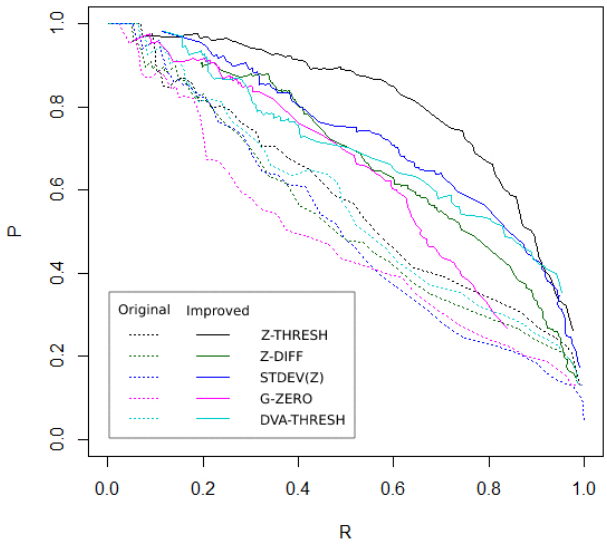

Figure 2. The precision-recall curve graphs, © 2019 IEEE [4].

Experimental results with the Z-THRESH algorithm on the three different datasets (the data collected by cars, the data collected by the scooter, and the aggregated data) showed that not only did it provide better results, but the improved algorithm also showed more stable results when changing vehicles. This proved that the application of the Grubbs test allowed the algorithm to adapt well to real conditions because the amplitude of the acceleration data when using cars and scooters was very different.

The Precision-recall curves in Figure 3 show that the original Z-THRESH algorithm resulted in a significant reduction of the precision when the data became diverse. Meanwhile, the improved algorithm gave good and very stable results, as all three curves were close to each other. Analyzing the F-measure graph (see Figure 4), one can also see that with the original Z-THRESH algorithm, the F-measure reached a maximum at very different threshold values when testing on datasets. In contrast, with the improved algorithm, not only the maximum value of the F-measure was greater,

but more importantly, the F-measure reached a similar maximum value even when it was applied to different datasets.

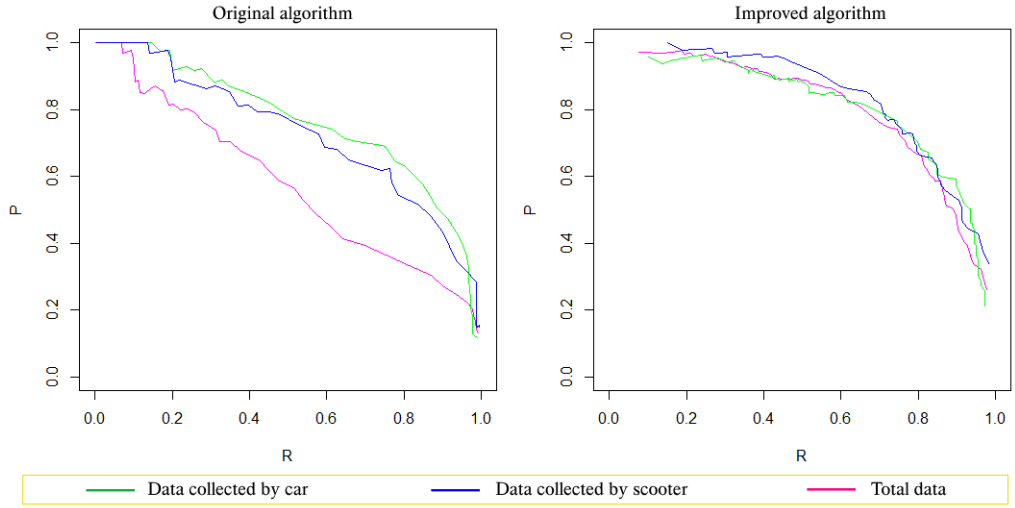

Figure 3. The precision-recall curves graphs of the Z-THRESH algorithm on the three datasets ©, 2019 IEEE [4].

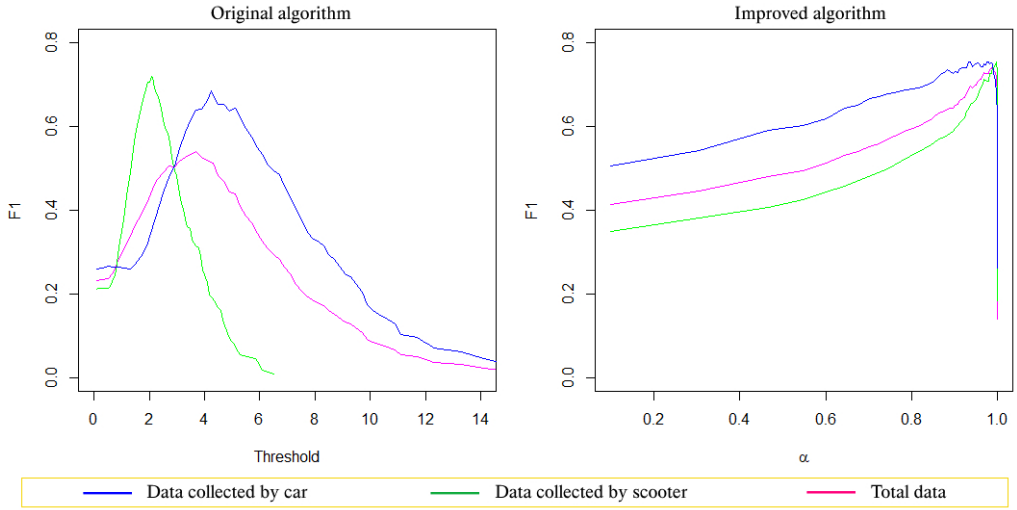

Figure 4. The F-measure graphs of the Z-THRESH algorithm on the three datasets, © 2019 IEEE [4].

We applied and tested the application of test outliers on the detection of road anomalies. The results showed that this approach achieved very good results. The principle of finding anomalies compared to the surrounding data can also be applied to other types of data and other applications. For example, the application of a fall detection on acceleration data also has approaches based on a threshold [17] that can be improved.

4. Anomaly Identifier

When multiple anomalies are detected and sent to the center from multiple smartphones, an anomaly can be reported several times. The position of an anomaly was reported determined by GPS, and there were some errors. The problem was aggregating that data to determine the more accurate location and reliability of the information about the anomaly.

There have been some research works discussing anomaly data aggregation methods. Notably, Jakob Eriksson et al. [6] proposed a clustering method whereby two anomalies were grouped in a cluster if their distance was less than a given value Δd. However, the maximum distance between

anomalies in a cluster was limited by a given value Δt. According to the authors, an event was valid if there was at least k elements in the cluster to which it belonged. Zhaojian Li et al. [18] proposed a clustering method for grouping anomalies based on the Mahalanobis distance [19]. The main idea for clustering was as follows: an event with coordinates x was set in cluster C if the Mahalanobis distance $D(x, C)$ was less than a given p value. The Mahalanobis distance was computed using the equation:

$$D(x, C) = \sqrt{(x - \mu_C)^{\mathrm{T}} \Sigma_C^{-1} (x - \mu_C)}$$

where μ_C is the weighted mean and Σ_C is the weighted covariance matrix of C. The authors proposed to use the event time as a parameter to compute μ_C and Σ_C, i.e.,

$$\mu_C = \frac{\sum_{k=1}^{K_C} f(t - t_{Ck}) x_{Ck}}{\sum_{k=1}^{K_C} f(t - t_{Ck})}$$

$$\Sigma_C = \frac{\sum_{k=1}^{K_C} f(t - t_{Ck})(x_{Ck} - \mu_C)(x_{Ck} - \mu_C)^{\mathrm{T}}}{\sum_{k=1}^{K_C} f(t - t_{Ck})}$$

where t_{Ck} and x_{Ck} are the time and coordinates of data point k^{th} in cluster C and t is the current time. Function f is an exponential function of the form $f(\tau) = \alpha^{-\lambda\tau}$ where $\alpha > 1$ and $\lambda > 0$ are two positive scalars.

The common disadvantage of the above two methods was that they only stopped at the clustering, not yet calculating the position of anomalies, as well as the lack of reliability assessment. In addition, Jakob Eriksson's method was unclear about how to remove the anomalies from a cluster if the maximum distance in the cluster exceeded the allowed value. Therefore, we propose a method of anomaly position aggregation in which the main part is the mean shift-based algorithm.

The position of a reported anomaly is a GPS location with horizontal accuracy. GPS location data can be considered Gaussian distributed around the real position [20–22], and the horizontal accuracy of the GPS on the smartphone is the RMS (Root Mean Squared) accuracy in two dimensions [22], i.e.,

$$RMS = \sqrt{\sigma_x^2 + \sigma_y^2}$$

where σ_x^2 and σ_y^2 are the variance according to the axes. As GPS data on smartphones do not contain specific σ_x and σ_y values, one can approximate σ_x and σ_y to $accuracy/\sqrt{2}$. In the scope of this study, we used a variance value $\sigma^2 = \sigma_x^2 = \sigma_y^2$ for each data point. In addition, the calculation of the GPS error was based on the assumption that the components of the error were independent [23], so we assumed that the GPS position on the plane was distributed according to the Gaussian distribution with the covariance matrix $\Sigma = \mathrm{diag}(\sigma^2, \sigma^2)$. In practice, the type of GPS accuracy should be determined as a parameter to adjust the way of computing the variance.

The anomaly locations looked like the observations of a bi-variate Gaussian mixture model in which variances corresponding to each observation point were known. Our solution was to use the mean shift method to find the highest points of the probability density estimation function. However, as the location of the anomalies is scattered throughout the world, we first needed to split the data into non-interconnected clusters, to increase accuracy and reduce computation time for the mean shift-based algorithm.

4.1. Simple Clustering

This clustering is a pretreatment to divide the reported anomalies into isolated regions. Assuming p_i and p_j are two points corresponding to reported anomalies, these two points are referred to as *far apart* if the distance between them is greater than a specific value d_{ij}:

$$\text{far}(p_i, p_j) = \begin{cases} \text{true} & \text{if } \|p_i - p_j\| > d_{ij} \\ \text{false} & \text{otherwise} \end{cases}$$

where σ_i and σ_j are the standard deviations associated with points p_i and p_j, respectively. Then, let function $include(C, p)$ used to determine if point p belongs to cluster C be defined by:

$$\text{include}(C, p) = \begin{cases} \text{true} & \text{if } \exists\, q \in C \ / \ \text{far}(p,q) = \text{false} \\ \text{false} & \text{otherwise} \end{cases}$$

The value d_{ij} needs to be chosen large enough so that in the case where p_i and p_j are not in the same cluster, the probability that these two points belong to the same real anomaly is very small. Note that under the Gaussian distribution assumption mentioned earlier, the probability of an observation falling outside the circle with radius r and the center being the anomaly position is $\exp\left(-\frac{r^2}{2\sigma^2}\right)$, where σ^2 is the variance. Suppose the number of observations is n, then the probability of having exactly one observation falling outside this circle, according to the binomial distribution, is $C_n^1 \exp\left(-\frac{r^2}{2\sigma^2}\right)^1 \left(1 - \exp\left(-\frac{r^2}{2\sigma^2}\right)\right)^{n-1} < n * \exp\left(-\frac{r^2}{2\sigma^2}\right)$. Hence, if $p_i - p_j > d_{ij}$ and $p_j \in C$, the probability of p_i belonging to an anomaly of cluster C will be less than $\delta = N_{max} * \exp\left(-\frac{d_{ij}^2}{2\sigma_i^2}\right)$, where N_{max} is the maximum number of reports possible for an anomaly. The value d_{ij} can be selected based on the desired value of δ. In fact, we also need to ensure that p_j does not belong to an anomaly of the cluster of p_i. Therefore, we propose the following formula:

$$d_{ij} = \max(\sigma_i, \sigma_j)\sqrt{-2\ln\left(\frac{\delta}{N_{\max}}\right)} \tag{1}$$

For example, if $N_{max} = 10{,}000$, $\delta = 0.0001$, then $d_{ij} \approx 6 * max(\sigma_i, \sigma_j)$. The value N_{max} can be predicted from reported data. Observe that it is updated, but does not grow infinitely over time since the system removes old reports periodically.

Formula (1) only ensures that the value of d_{ij} is "large enough", not optimal for clustering. The goal is for this simple clustering to be performed quickly in a cumulative manner. More detailed clustering technique is addressed in Section 4.2.

Algorithm 1 Update clusters.

INPUT: $C(C_1, C_2, \ldots, C_K)$ {current cluster list}, p {new data point}
OUTPUT: C {new cluster list}

```
C* ← ∅ {C* is to contain all current clusters that p belongs to}
for k ∈ {1, ..., K} do
    if include(C_k, p) then
        C* ← C* ∪ {C_k}
    end if
end for
{Next, take the union of all clusters in C* and add p to get the new cluster}
C ← C \ C*
C_new ← (⋃_{i=1}^{|C*|} C_i*) ∪ {p}
C ← C ∪ {C_new}
```

The data clustering algorithm is executed cumulatively, i.e., it is called when a new data point is added. The algorithm to update the cluster is presented in Algorithm 1.

4.2. Mean Shift-Based Algorithm to Find Anomaly Positions

The next step after applying the simple clustering method presented above is to find the local maxima of the probability density (modes) based on the reported anomaly positions and the associated GPS accuracy. A mode is the most likely point where anomalies have been detected. Given the characteristic of the reported anomalies data, we found that the *variable bandwidth mean shift* method with the *Gaussian kernel* was an effective option to solve the problem. Dorin Comaniciu showed that mean shift to variable bandwidth is an adaptive estimator of the normalized gradient of the underlying density [24]. Carreira-Perpinan and Miguel A also confirmed that, when the kernel was Gaussian, mean shift was an expectation-maximization algorithm [25]. As far as the problem of the aggregation of anomalies is concerned, it can be seen that it was a Gaussian mixture model. Computations via this model can be quite complicated, and each observation corresponds to a different variance. Nevertheless, this complexity can be handled by a variable bandwidth mean shift in which each data point is combined with a bandwidth value. Moreover, the mean shift algorithm solves two problems simultaneously: the mode-finding problem to locate anomalies and the clustering problem to calculate the reliability of the identified modes.

Let $p_i, i = 1 \ldots n$ be a set of points in the $\mathcal{R}^d$ space and the associated bandwidth h_i, the density estimator computed at a given point p with kernel profile $k(p)$ is given by [26]:

$$\hat{f}(p) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h_i^d} k \left(\left\| \frac{p - p_i}{h_i} \right\|^2 \right) \tag{2}$$

The local maximum of the density function that can be iteratively reached using a mean shift vector is given by [26]:

$$m(p) = \frac{\sum_{i=1}^{n} \frac{p_i}{h_i^{d+2}} g\left(\left\| \frac{p-p_i}{h_i} \right\|^2 \right)}{\sum_{i=1}^{n} \frac{1}{h_i^{d+2}} g\left(\left\| \frac{p-p_i}{h_i} \right\|^2 \right)} - p \tag{3}$$

where $g(x) = -k'(x)$. We use the multivariate Gaussian kernel:

$$k(x) = \exp\left(-\frac{1}{2}x\right) \tag{4}$$

so that:

$$g(x) = -\frac{1}{2} \exp\left(-\frac{1}{2}x\right) \tag{5}$$

In this study, each point p_i was considered an observation of a Gaussian distribution with the associated standard deviation σ_i. We propose to use bandwidth $h_i = \lambda\sigma_i$, with λ being an experimental parameter adjusted around one to get the best results for each situation. The anomalies were considered in the plane, and after replacing $d = 2$, $h_i = \lambda\sigma_i$ and Equation (5) in Equation (3), we get:

$$m(p) = \frac{\sum_{i=1}^{n} \frac{p_i}{\lambda^4 \sigma_i^4} \exp\left(-\frac{1}{2} \left\| \frac{p-p_i}{\lambda\sigma_i} \right\|^2\right)}{\sum_{i=1}^{n} \frac{1}{\lambda^4 \sigma_i^4} \exp\left(-\frac{1}{2} \left\| \frac{p-p_i}{\lambda\sigma_i} \right\|^2\right)} - p \tag{6}$$

Based on the mean shift vector, we developed an algorithm that is similar to the one presented in [27] to find the maximum density points (see Algorithm 2). The algorithm assigns a weight for every maximum point found. These weighted points are stored in the system to assess reliability. We dealt

with it as an anomaly if the associated weight was greater than a given threshold. It is also possible to improve the algorithm by eliminating mode points that follow some weight-related criteria.

Algorithm 2 Anomaly identification.

INPUT:
$p_1, p_2, \ldots, p_n$ {list of data points}
$\sigma_1, \sigma_2, \ldots, \sigma_n$ {list of standard deviations corresponding to points}
λ {bandwidth parameter}
ε, ρ {error parameters}
OUTPUT: Q, w {list of modes and list of weights}
$Q \leftarrow \emptyset$
for $i \in \{1, \ldots, n\}$ **do**
$p \leftarrow p_i$
repeat
$$m(p) \leftarrow \frac{\sum_{j=1}^{n} \frac{p_j}{\lambda^4 \sigma_j^4} \exp(-\frac{1}{2}\left\|\frac{p-p_j}{\lambda\sigma_j}\right\|^2)}{\sum_{j=1}^{n} \frac{1}{\lambda^4 \sigma_j^4} \exp(-\frac{1}{2}\left\|\frac{p-p_j}{\lambda\sigma_j}\right\|^2)} - p$$
$p \leftarrow p + m(p)$
until $|m(p)| < \varepsilon$
$q = nearestPoint(Q, p)$ {*nearestPoint(Q,p)* returns a point in Q that is closest to p}
if $\exists q$ **and** $\|q - p\| < \rho$ **then**
$w(q) \leftarrow w(q) + \frac{1}{\sigma_i^2} \exp\left(-\frac{\|q-p_i\|^2}{2\sigma_i^2}\right)$
else
$w(p) \leftarrow \frac{1}{\sigma_i^2} \exp\left(-\frac{\|p-p_i\|^2}{2\sigma_i^2}\right)$
$Q \leftarrow Q \cup \{p\}$
end if
end for

We recommend using weight function $w(p)$ as a proportion of $\hat{f}(p)$ with bandwidth σ_i:

$$\mathrm{w}(p) = \sum_{i=1}^{n} \frac{1}{\sigma_i^2} \exp\left(-\frac{\|p - p_i\|^2}{2\sigma_i^2}\right) \tag{7}$$

To understand the properties of function w, we first consider the function v as follows:

$$\mathrm{v}(p) = \sum_{i=1}^{n} \exp\left(-\frac{\|p - p_i\|^2}{2\sigma_i^2}\right) \tag{8}$$

Assume there is an anomaly at point $p(0,0)$ and there is an observation of the anomaly at point $q(x,y)$. According to the Gaussian distribution with variance $\sigma_x^2 = \sigma_y^2 = \sigma^2$, the expected value of the $v(p)$ functions is given by:

$$\begin{aligned} \mathrm{E}[\mathrm{v}(\mathrm{p})] &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \mathrm{v}(\mathrm{p}) f(x,y)\,dxdy = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)\frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)dxdy \\ &= \frac{1}{2\pi\sigma^2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)dxdy \end{aligned} \tag{9}$$

where $f(x,y)$ is the probability density function of a bivariate normal distribution in the case of independent x and y. The integral in Equation (9) can be computed in polar coordinates, i.e.:

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \exp\left(-\frac{x^2+y^2}{\sigma^2}\right)dxdy = \int_{0}^{+\infty}\int_{0}^{2\pi} \exp(-\frac{r^2}{\sigma^2})\, r\, dr\, d\theta = \pi\sigma^2$$

This leads to an expected value of function v equal to:

$$\mathrm{E[v(p)]} = \frac{1}{2} \tag{10}$$

From Equations (7), (8), and (10), the expected value of function w is:

$$\mathrm{E[w(p)]} = \sum_{i=1}^{n} \frac{\mathrm{E[v(p)]}}{\sigma_i^2} = \sum_{i=1}^{n} \frac{1}{2\sigma_i^2} = \frac{n}{2}\left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sigma_i^2}\right) = n\frac{\overline{(\sigma_i^{-2})}}{2} \tag{11}$$

Equation (11) shows that the expected value of the function w is directly proportional to the number of reports. Moreover, the smaller the reporting error, the bigger the value in w. As a result, function w can be used as a weighted function to evaluate the reliability of an anomaly.

Note that Equation (11) applies only for the case of one anomaly. In practice, there might be multiple anomalies, giving rise to several modes. Only points in the *attraction basin* of a mode are computed in the weight function of that mode. The attraction basin of a mode is the largest region in which all the points have trajectories (defined by the mean shift vector) that lead to this mode [28].

In Algorithm 2, two location error parameters (ε and ρ) are used. The `repeat ...until` loop stops when the mean shift vector is smaller than a positive tolerance [27]. Tolerance value ε is chosen according to the specific application. The greater ε, the faster the program. However, the accuracy of the location of the modes is reduced. With this stop condition, the program can give several results around a real mode. To fix this, a ρ parameter is added. The algorithm must ensure:

(1) the distance between any two modes is greater than or equal to ρ;
(2) if a potential point is removed, there must be at least one other convergence point selected as the mode so that the distance between them is smaller than ρ.

The ρ parameter must be large enough to ensure that the program only produces one point for each mode, and it must be small enough to prevent different real modes from merging into a single one. In our simulation experiments, we first chose ε and then did an experiment to choose a suitable ρ.

4.3. Simulation and Results

To prove the effectiveness of Algorithm 2 and consider the dependence of the result on the λ parameter, we conducted the simulation as follows. From a set of chosen points $P = \{p_1, p_2, \ldots, p_r\}$ representing the anomaly positions, the simulation program generated the test dataset this way: from each point p_i, create k points $P_i = \{p_{i1}, p_{i2}, \ldots, p_{ik}\}$ that represent the observations reported for anomaly i. Points p_{ij} are randomly generated according to the Gaussian distribution with variance $\sigma_{xij}^2 = \sigma_{yij}^2 = \sigma_{ij}^2$ around point p_i. The value of σ_{ij} is also a random value between σ_{min} and σ_{max}. In these simulations, σ_{min} was set to 2.12 and σ_{max} was set to 8.48, which corresponds to a GPS accuracy between 3 m and 12 m, respectively.

The set of $k \times r$ generated points served as the test data. The list of obtained maximum points was compared with the original set of points P to evaluate the results.

During the experiment, we first observed the effect of the parameter on the results in order to select the parameters before proceeding to obtain the final results. We experimented with the case of one mode, $\lambda = 0.9$, $\varepsilon = 0.01$, with the respective number of points of 20, 50, 100, and 1000. The program-produced convergence points with the average distance and the standard deviation,

calculated over 1000 tests, were respectively $(0.008, 0.0123)$, $(0.012, 0.0147)$, $(0.014, 0.0154)$, and $(0.013, 0.0078)$. We choose the parameter $\rho = 0.1$, which is a sufficiently large value, to ensure that these points were only counted for one mode. On the other hand, this value was very small compared to the GPS error. Hence, we chose $\varepsilon = 0.01$ and $\rho = 0.1$ for the experiments.

Our experiments were conducted on a set of anomalies P consisting of 10 points of imaginary anomalies. The center of the coordinate system was at point $(lat = 48.817456, lon = 2.419799)$. The vertical axis was in the south-north direction. The horizontal axis was directed from west to east. The length unit was meters. The ten points were at coordinates (−23,0), (−28,0), (−8,0), (0,0), (15,0), (25,0), (37,0), (0,35), (−5,20), and (10,22).

In the first experiment, we studied the results with different values for λ. For each of the ten original points in P, we generated k = 50 data points. Thus, the cluster reported 500 anomalies. Experimental results on this dataset, with the corresponding parameters λ, were 0.4, 0.6, 0.9, and 1.6, respectively, as shown in Figure 5. These results showed that when the bandwidth was too small, the mean shift algorithm obtained too many maxima points. Then, the larger the bandwidth, the smaller the number of maxima points as the nearest maxima points gradually became one. With the appropriate λ values, e.g., $\lambda = 0.9$ in the above experiment, the results were very good, except for two very close points. In the above experiments, they were located at a 5-m distance, and the program only identified them as a single point located in the middle.

We conducted the next experiment to evaluate quantitatively the dependence of the algorithm results on λ. First, we need to define an objective function. Assume P is the set of initial anomalies, Q is the set of modes obtained from the algorithm, $p \in P$, and $q \in Q$. Let:

$$\text{found}(p, Q) = \begin{cases} \text{true} & \text{if } \exists\, q^* \in Q \text{ such that } \|p - q^*\| < \omega \\ \text{false} & \text{otherwise} \end{cases}$$

$$\text{matched}(q, P) = \begin{cases} \text{true} & \text{if } \exists\, p^* \in P \text{ such that } \|p^* - q\| < \omega \\ \text{false} & \text{otherwise} \end{cases}$$

Then, objective function T can be defined as:

$$T(P, Q) = |\{p : p \in P \wedge \text{found}(p, Q)\}| - |\{q : q \in Q \wedge \neg\, \text{matched}(q, P)\}|$$

We chose ω = 4 m for these experiments. We conducted experiments with sets of two points and determined that, if the distance between the two points was less than 4 m, the probability of identifying both anomalies was less than 10%.

Figure 6 shows the results of the experiments for the 10 original points above with six different values for k. For each value of k, the test program generated 100 datasets, each of which was used to test each value of λ from 0.5–1.5 (with steps of 0.025). The graphs allowed us to draw some of the following properties:

- In general, the larger the number of data, the better the results. With $k \geq 50$, the result of the program was very good, i.e., close to 10 (as data were generated with 10 maxima to find).
- The larger the dataset, the smaller the required λ value for the algorithm to achieve the optimal result (see Figure 6b).
- More experiments on real data were needed to estimate good λ values as a function of the total number of data points n. According to past experiments, λ should be chosen in the range from 0.7–0.9. Since this value is quite small, the program usually results in identifying more anomalies than there are effectively, especially when the size of the dataset is small. However, the program tends to eliminate weighted points below a given threshold. This not only eliminates cases where there are few reported data, but it also allows the removal of small maxima points, often located far from real anomalies.

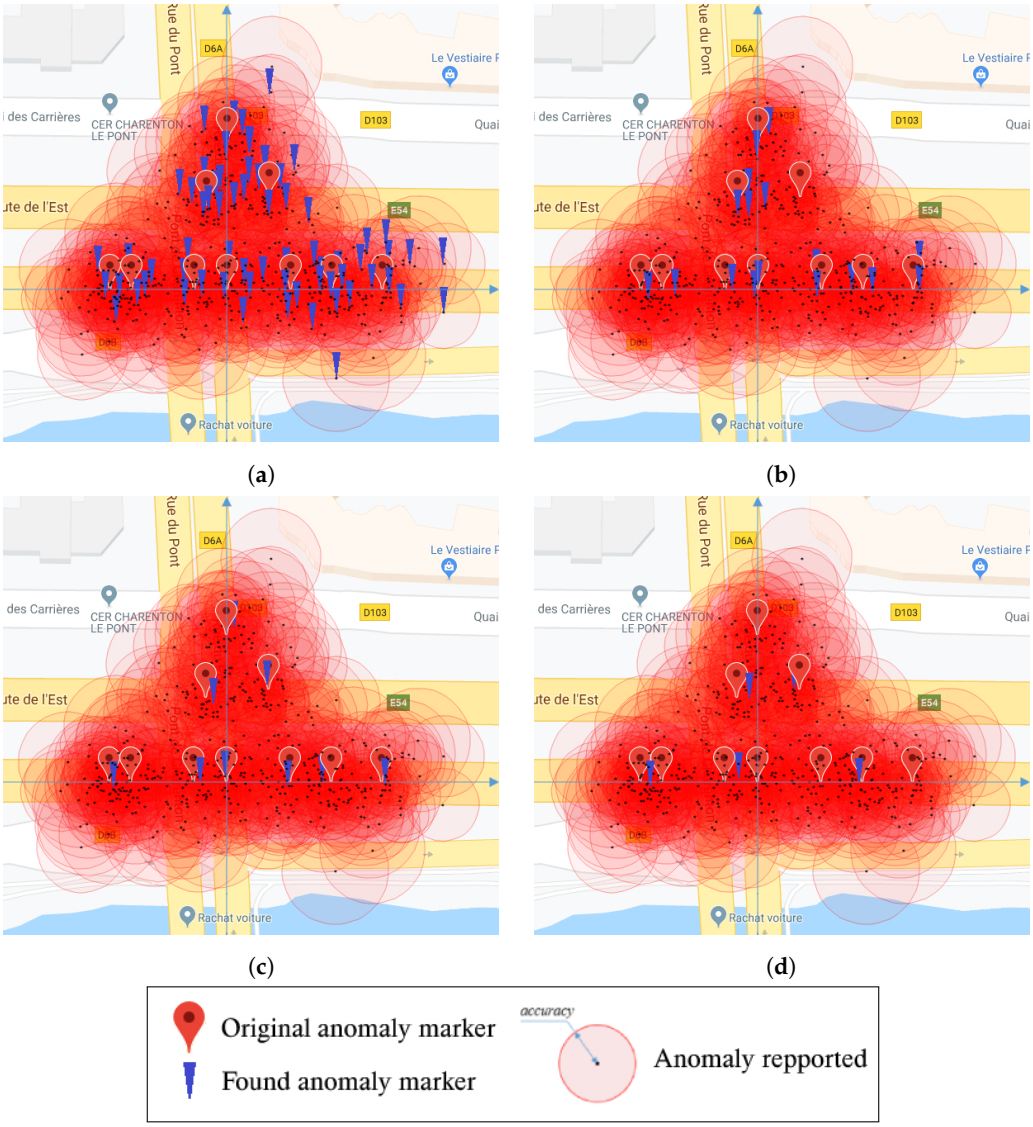

Figure 5. Visual results of the simulation. (**a**) $\lambda = 0.4$; (**b**) $\lambda = 0.6$; (**c**) $\lambda = 0.9$; (**d**) $\lambda = 1.6$.

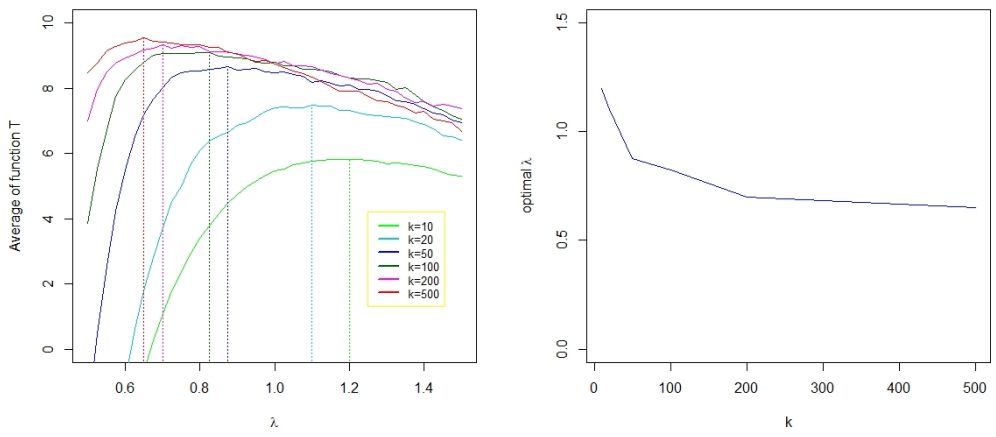

Figure 6. Objective function and optimal λ. (**a**) Average of the objective function according to λ; (**b**) optimal value of λ according to k.

5. Conclusions and Future Work

In this article, we proposed an architecture to detect road anomalies using a mobile phone sensor system. For this model, we focused on building lightweight algorithms to be used on smartphones to detect anomalies and on building algorithms to aggregate anomaly data on the server.

For the anomaly detection, we proposed to apply the Grubbs test to improve the classic threshold-based methods chosen for their lightweight and adaptive real-time approach. Experiments for road anomaly detection on real data collected by both cars and scooters suggested that the method we proposed gave good results and was well adapted to different types of vehicles.

Regarding the anomaly data aggregation, we proposed a two-stage method. First, reported anomaly data were divided into separate clusters. Then, a mean shift-based algorithm was applied to find modes, i.e., the most likely position of the anomalies. Simulations showed very good results. This experiment also allowed analyzing how to select the appropriate parameters according to the data characteristics.

In the future, we aim at improving the model by applying a machine learning method in the server to increase the accuracy of the anomaly detection and classify them. We will also do experiments on real data to validate the anomaly data aggregation method.

Author Contributions: Conceptualization, V.K.N., É.R. and R.M.; Data curation, V.K.N.; Formal analysis, É.R.; Methodology, V.K.N., É.R. and R.M.; Software, V.K.N.; Supervision, É.R.; Validation, É.R.; Writing—original draft, V.K.N.; Writing—review & editing, É.R. and R.M.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SVM	Support Vector Machines
FFT	Fast Fourier Transformation
ROC	Receiver Operating Characteristic curve
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative

References

1. Arora, S.; Venkataraman, V.; Zhan, A.; Donohue, S.; Biglan, K.M.; Dorsey, E.R.; Little, M.A. Detecting and monitoring the symptoms of Parkinson's disease using smartphones: A pilot study. *Park. Relat. Disord.* **2015**, *21*, 650–653. [CrossRef] [PubMed]
2. Habib, M.; Mohktar, M.; Kamaruzzaman, S.; Lim, K.; Pin, T.; Ibrahim, F. Smartphone-based solutions for fall detection and prevention: Challenges and open issues. *Sensors* **2014**, *14*, 7181–7208. [CrossRef]
3. Kanhere, S.S. Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces. In Proceedings of the 2011 IEEE 12th International Conference on Mobile Data Management, Lulea, Sweden, 6–9 June 2011; Volume 2, pp. 3–6.
4. Van Khang, N.; Renault, É. Cooperative Sensing and Analysis for a Smart Pothole Detection. In Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 1785–1790.
5. Nguyen, V.K.; Renault, É.; Ha, V.H. Road Anomaly Detection Using Smartphone: A Brief Analysis. In Proceedings of the International Conference on Mobile, Secure, and Programmable Networking, Paris, France, 18–20 June 2018; pp. 86–97.
6. Eriksson, J.; Girod, L.; Hull, B.; Newton, R.; Madden, S.; Balakrishnan, H. The pothole patrol: Using a mobile sensor network for road surface monitoring. In Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services, Breckenridge, CO, USA, 17–20 June 2008; pp. 29–39.

7. Mohan, P.; Padmanabhan, V.N.; Ramjee, R. Nericell: Rich monitoring of road and traffic conditions using mobile smartphones. In Proceedings of the 6th ACM conference on Embedded Network Sensor Systems, Raleigh, NC, USA, 5–7 November 2008; pp. 323–336.
8. Vittorio, A.; Rosolino, V.; Teresa, I.; Vittoria, C.M.; Vincenzo, P.G. Automated sensing system for monitoring of road surface quality by mobile devices. *Procedia Soc. Behav. Sci.* **2014**, *111*, 242–251. [CrossRef]
9. Mednis, A.; Strazdins, G.; Zviedris, R.; Kanonirs, G.; Selavo, L. Real time pothole detection using android smartphones with accelerometers. In Proceedings of the 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS), Barcelona, Spain, 27–29 June 2011; pp. 1–6.
10. Perttunen, M.; Mazhelis, O.; Cong, F.; Kauppila, M.; Leppänen, T.; Kantola, J.; Collin, J.; Pirttikangas, S.; Haverinen, J.; Ristaniemi, T.; et al. Distributed road surface condition monitoring using mobile phones. In Proceedings of the International Conference on Ubiquitous Intelligence and Computing, Banff, AB, Canada, 2–4 September 2011; pp. 64–78.
11. Seraj, F.; van der Zwaag, B.J.; Dilo, A.; Luarasi, T.; Havinga, P. RoADS: A road pavement monitoring system for anomaly detection using smart phones. In *Big Data Analytics in the Social and Ubiquitous Context*; Springer: Cham, Switzerland, 2014; pp. 128–146.
12. Bhoraskar, R.; Vankadhara, N.; Raman, B.; Kulkarni, P. Wolverine: Traffic and road condition estimation using smartphone sensors. In Proceedings of the 2012 Fourth International Conference on Communication Systems and Networks (COMSNETS), Bangalore, India, 3–7 January 2012; pp. 1–6.
13. Grubbs, F.E. Procedures for detecting outlying observations in samples. *Technometrics* **1969**, *11*, 1–21. [CrossRef]
14. Garcia, F. *Tests to Identify Outliers in Data Series, Pontifical Catholic University of Rio de Janeiro*; Industrial Engineering Department: Rio de Janeiro, Brazil, 2012.
15. Powers, D.M. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation. 2011. Available online: http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A635335&dswid=-1937 (accessed on 15 August 2011).
16. Goutte, C.; Gaussier, E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In Proceedings of the European Conference on Information Retrieval, Santiago de Compostela, Spain, 21–23 March 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 345–359.
17. Mubashir, M.; Shao, L.; Seed, L. A survey on fall detection: Principles and approaches. *Neurocomputing* **2013**, *100*, 144–152. [CrossRef]
18. Li, Z.; Filev, D.P.; Kolmanovsky, I.; Atkins, E.; Lu, J. A new clustering algorithm for processing GPS-based road anomaly reports with a mahalanobis distance. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1980–1988. [CrossRef]
19. Mahalanobis, P.C. *On the Generalized Distance in Statistics*; National Institute of Science of India: Bangalore, India, 1936.
20. Tiberius, C.; Borre, K. Are GPS data normally distributed. In *Geodesy Beyond 2000*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 243–248.
21. Johansson, M. Estimering av GPS Pålitlighet och GPS/INS Fusion. 2013. Available online: http://www.bioinfo.in/journalcontent.php?vol_id=115&id=51&month=12&year=2011 (accessed on 20 June 2019).
22. Drane, C.; Macnaughtan, M.; Scott, C. Positioning GSM telephones. *IEEE Commun. Mag.* **1998**, *36*, 46–54. [CrossRef]
23. Langley, R.B. Dilution of precision. *GPS World* **1999**, *10*, 52–59.
24. Comaniciu, D. An algorithm for data-driven bandwidth selection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 281–288. [CrossRef]
25. Carreira-Perpinan, M.A. Gaussian mean-shift is an EM algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 767–776. [CrossRef] [PubMed]
26. Ramesh, D.C.V.; Meer, P. The variable bandwidth mean shift and data-driven scale selection. In Proceedings of the Eigth International Conference on Computer Vision, Vancouver, BC, Canada, 7–14 July 2001; pp. 438–445.

27. Carreira-Perpinán, M.A. A review of mean-shift algorithms for clustering. *arXiv* **2015**, arXiv:1503.00687.
28. Ozertem, U.; Erdogmus, D.; Jenssen, R. Mean shift spectral clustering. *Pattern Recognit.* **2008**, *41*, 1924–1938. [CrossRef]