Hindawi
Security and Communication Networks
Volume 2020, Article ID 8858444, 15 pages
https://doi.org/10.1155/2020/8858444

Research Article

A Framework of Abnormal Behavior Detection and Classification Based on Big Trajectory Data for Mobile Networks

Haiyan Zhang,[1,2] **Yonglong Luo**,[1,2] **Qingying Yu**,[1,2] **Liping Sun**,[1,2] **Xuejing Li**,[1,2] **and Zhenqiang Sun**[1,2]

[1]*School of Computer and Information, Anhui Normal University, 241002 Wuhu, Anhui, China*
[2]*Anhui Provincial Key Laboratory of Network and Information Security, 241002 Wuhu, Anhui, China*

Correspondence should be addressed to Yonglong Luo; ylluo@ustc.edu.cn

Received 28 September 2020; Revised 8 November 2020; Accepted 24 November 2020; Published 22 December 2020

Academic Editor: Zhe-Li Liu

Big trajectory data feature analysis for mobile networks is a popular big data analysis task. Due to the large coverage and complexity of the mobile networks, it is difficult to define and detect anomalies in urban motion behavior. Some existing methods are not suitable for the detection of abnormal urban vehicle trajectories because they use the limited single detection techniques, such as determining the common patterns. In this study, we propose a framework for urban trajectory modeling and anomaly detection. Our framework takes into account the fact that anomalous behavior manifests the overall shape of unusual locations and trajectories in the spatial domain as well as the way these locations appear. Therefore, this study determines the peripheral features required for anomaly detection, including spatial location, sequence, and behavioral features. Then, we explore sports behaviors from the three types of features and build a taxi trajectory model for anomaly detection. Anomaly detection, including sports behaviors, are (i) detour behavior detection using an algorithm for global router anomaly detection of trajectories having a pair of same starting and ending points; this method is based on the isolation forest algorithm; (ii) local speed anomaly detection based on the DBSCAN algorithm; and (iii) local shape anomaly detection based on the local outlier factor algorithm. Using a real-life dataset, we demonstrate the effectiveness of our methods in detecting outliers. Furthermore, experiments show that the proposed algorithms perform better than the classical algorithm in terms of high accuracy and recall rate; thus, the proposed methods can accurately detect drivers' abnormal behavior.

1. Introduction

Big data analysis is the detection of massive data and a type of thinking process, technology, and resource. The trajectory data for the mobile networks, which is a branch of big data, comprise a rich sequence of geospatial locations with timestamps and carry the information of the moving object's actual movement. They have the characteristics of time and space, spatially static but temporally dynamic [1]. A massive amount of vehicle trajectory data is collected by GPS-embedded vehicles. The "big trajectory data" under the mobile networks have contributed to the emergence of many data-driven trajectory-based applications such as route recommendation [2], transit time estimation [3, 4], traffic dynamic analysis [5], fraud detection [6], and city planning [7]. Analysis on such data to serve fields including intelligent transportation and smart cities has attracted the interest of a large number of researchers [8].

An abnormality generally implies that a data object is extremely deviant from the remaining set of retrieved data due to some of its unusual features. An abnormal trajectory differs clearly from most trajectories scrutinized under a similarity evaluation mechanism. An obvious rare pattern may indicate an abnormal event [9]. The detection results can help identify suspicious activities of vehicles and be used in many applications such as security surveillance, scheduling, and city planning [7, 10]. In the surveillance application, vehicle trajectories can be used in automatic visual surveillance [11], traffic management [12], suspicious activity detection [13], sports video analysis [14], video

summarization [14], synopsis generation [15], and video-to-text descriptors [16], among others. Ngan et al. [17] adopted a Dirichlet process mixture model (DPMM) for detecting outliers in large-scale urban traffic data. Kingan and Westhuis [18] presented a regression model approach for average daily traffic. Therefore, outlier detection is an important analysis task.

Every day, thousands of people are victims of traffic accidents, which are generally directly related to driver behavior. According to the World Health Organization, the total number of road traffic deaths worldwide is approximately 1.3 million per year. The primary causes of accidents include speeding, drunk driving, unsafe lane switching, and incorrect turns, among others [19, 20]. Therefore, we divide anomalous trajectories into three categories: (1) global router anomaly, (2) local speed anomaly, and (3) local shape anomaly. Specifically, the global router anomaly means that the driver adds extra travel for some reasons. This anomaly occurs in various instances; for example, the taxi driver fraudulently increases the itinerary of the customer to obtain additional benefits, or the driver chooses another route to avoid traffic congestions or road repairs. Local speed anomaly refers to the vehicle exceeding the speed limitation specified for some road sections, such as roads near schools and hospitals. Local shape anomaly refers to the local shape of the trajectory meandering. This anomaly can occur due to several reasons such as drunk driving, in case of which the alcohol content in the blood of the vehicle driver exceeds a specified limit, leading to the slowing down of the reflex arc nerve and decrease in the tactile ability of the driver; these problems in turn cause instability of the steering wheel, eventually forcing the driver to continuously change lanes during driving.

To address the aforementioned problems, this study proposes three abnormal trajectory detection algorithms. The main contributions of this study are summarized as follows:

(1) This study systematically analyzes the abnormal behaviors of taxis, including detour behavior, speed anomaly, and local shape anomaly. According to the different anomalies causing abnormal taxi trajectory, different solutions are proposed.

(2) Most people who are new to the city will choose to take a taxi. Considering the situation of taxi detours and planned delay and safety issues, this study proposes a novel global router anomaly detection algorithm. When the starting and ending points are the same, the trajectory that greatly differs from the conventional route is considered a detour anomaly.

(3) In road sections where provision to take pictures to detect speed limit violation is absent, in order to detect taxis travelling at abnormal speed, this study proposes a method for detecting speed abnormal trajectories on the basis of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm.

(4) Considering the situation of unstable driving direction of the taxi caused by drunk driving or incorrect turns, this study proposes a method for detecting local shape abnormity on the basis of the local outlier factor (LOF) algorithm.

(5) Considerable research on urban monitoring has been conducted by analyzing videos or images. The detection framework proposed in this study detects anomalies in urban traffic by analyzing big trajectory data for the mobile networks. Thus, our framework can reduce the cost of urban monitoring and provide services for smart cities.

The rest of the paper is organized as follows. Section 2 introduces existing related work on abnormal trajectory detection. Section 3 introduces the assumptions adopted in this paper. Taxi abnormal trajectory detection algorithms are detailed in Section 4. Experimental setup and results are presented in Section 5, and the conclusions are presented in Section 6.

2. Related Work

Several abnormal trajectory detection approaches have been developed so far and continue to be explored; the existing approaches can be mainly classified into four categories: classification-based methods [21–24], distance-based methods [25–30], density-based methods [26, 27, 31–34], and statistical methods [35, 36].

2.1. Classification-Based Methods. A classification-based approach is a supervised learning algorithm. Commonly used classification algorithms include logistic regression [23], k-nearest neighbor algorithm [21], decision tree [22], and support vector machine [24]. The basic concept of the classification-based method is to train the algorithm through existing training samples to obtain an optimal model and then use this model to map all the inputs to the corresponding outputs, thereby making simple judgments on the output to achieve the purpose of the classification. It is an efficient method to classify unknown data. Therefore, when supervised learning algorithms are used, the dataset is divided into three categories during the experiments: training, test, and verification sets. The training set learns a classifier through labeled data; the test set is used to evaluate the performance of the algorithm; and the verification set can obtain appropriate hyperparameters. Both training and verification sets of this method must be labeled, and the classification effect of the classifier depends on the training dataset. Trajectory anomaly detection is difficult owing to the lack of ground truth data. Many researchers use human experts to label training data. However, in some cases, labeling data is impossible or difficult, which makes classification algorithms unreliable. In addition, given that the anomalies in the evolutionary trajectory are usually unknown and time dependent, it is impossible to obtain training data covering all anomalous instances in practical applications. Therefore, the classification-based anomaly detection method is not suitable for online anomaly detection of trajectory flow.

2.2. Distance-Based Methods. In the distance-based method, the trajectory in the trajectory dataset with a long distance, such as the Euclidean distance, Manhattan distance, and dynamic time warping (DTW) distance, from most trajectories is regarded as abnormal [25, 26, 30]. Knorr et al. introduced the concept of distance-based trajectory anomaly detection and, by conducting validation experiments using a database, proved that the method based on this concept can process high-dimensional data [25]. Hence, scholars have developed novel schemes on the basis of distance-based trajectory anomaly detection. This method uses a partition detection framework to detect abnormal trajectories and Hausdorff distance to measure the distance between two subtrajectories [26, 29]. Recently, San Román et al. proposed an abnormal trajectory detection method based on context-aware distance [28], wherein human trajectories are detected by video surveillance systems. First, the appropriate representation of each trajectory is selected by the polar coordinates of the trajectory. Then, the context-aware distance between trajectories is determined by the angle difference, the Euclidean distance, and the weighted average of the number of points in each trajectory. Subsequently, the trajectory distance matrix formed uses an unsupervised learning method to extract cohorts (clusters) of trajectories. Finally, an outlier detection method is used to detect anomalous trajectories in each cluster. Although distance-based detection methods are suitable for high-dimensional data, they are computationally expensive and time consuming. Moreover, they only adjust the abnormal behavior of the trajectory itself based on location information, ignoring the trajectory that is obviously different from its temporal and spatial neighbors in terms of nonlocation information.

2.3. Density-Based Methods. In density-based methods, outliers are objects in low-density areas [31, 32, 34]. Breunig et al. [31] defined a local outlier factor (LOF), which depends on the degree of isolation of an object relative to the surrounding neighborhood and has many desirable properties. For example, due to the local approach, LOF can identify outliers in a dataset that would not be outliers in another area of the dataset. The LOF and DBSCAN are similar, so some scholars used DBSCAN to detect outliers. A spatiotemporal (ST)-outlier detection method based on DBSCAN was proposed by adding the time dimension to a scheme presented by Kut and Birant [33]. First, a modified DBSCAN clustering algorithm is run on the tested data with two main modifications: (1) to support the time aspect, the tree is traversed to determine the space and time neighborhood of any object in a given radius; (2) to identify outliers, the algorithm allocates density factors to each cluster and compares the average value of clustering with the new clustering, when the clustering has different densities. After clustering, the potential outliers are detected. Furthermore, by checking the spatial neighbors, the objects are verified to be spatial exception values. Subsequently, the temporal neighbors of the spatial outliers identified in the previous step are checked. If the eigenvalue of the spatial anomaly is not significantly different from its temporal neighbor, it is not an ST exception. Otherwise, it is confirmed as an ST-outlier. The proposed scheme adds a limitation of the sliding window in the time dimension [37] and then divides trajectory anomaly detection into two categories: detection of abnormal trajectory points (PN-outliers) and detection of the entire trajectory (TN-outliers). This approach improves the detection efficiency, but the accuracy of trajectory detection is reduced. The time and space complexity of density clustering-based methods are linear or close to linear, so the detection of outliers is highly effective. The difficulty lies in the choice of the number of clusters and the existence of abnormal points. Extremely different results or effects are produced by different cluster numbers. Furthermore, a coarse quality greatly affects the quality of the outliers generated, and each clustering model is only suitable for specific data types.

2.4. Statistical Methods. In a statistical method, trajectory points are first modeled by assuming that a certain distribution is obeyed. Then, an abnormality is determined by checking whether the trajectory complies with the distribution model of trajectory points. The most frequent assumptions are Gaussian distributions [35] and multivariate Gaussian distributions [36]. When sufficient data and prior knowledge exist, using statistical methods for outlier detection can be very effective and efficient. However, such methods rely on a pathognomonic distribution model obtained with the used dataset, and it is difficult to select the parameters of the model. At present, few scholars use this method for abnormal trajectory detection. At the same time, most statistics-based outlier detection techniques use a single attribute. The current important question is how to model multivariate data (with multiple attributes).

The aforementioned abnormal trajectory detection methods can only detect abnormal trajectories in the driving range and driving direction and do not combine the characteristics of the abnormal driving behavior. In this study, we determine the peripheral features required for anomaly detection, including spatial location, sequence, and behavioral features. Then, we explore sports behaviors from the three types of features and build a taxi trajectory model for anomaly detection. The model systematically analyzes abnormal behaviors of drivers, but detection of such abnormalities is difficult.

3. Problem Description and Related Definitions

3.1. Problem Description. In this study, a given road network is denoted as $G(V, E)$ and a given trajectory dataset is denoted as RTS; we design algorithms to determine abnormal taxi trajectories and determine to which category of abnormal behavior of taxi drivers the detected trajectory belongs.

3.2. Related Definition. In this section, we provide the formal definitions of the parameters required for the algorithm.

Definition 1 (road network). The road network, denoted as $G(V, E)$, is a directed graph, where V represents the node set

(i.e., the starting point and the ending point of the road section) and E is the edge set (i.e., the road section). For road $e \in E$, $e.s \in V$ is the starting point of the road section and $e.d \in V$ is the ending point of the road section.

Definition 2 (free trajectory point). Let t be a timestamp and (x, y) be a location in $\Re^2$. A free trajectory point is defined as a triple (x, y, t), implying that an object is at location (x, y) at time t.

Definition 3 (restrained trajectory point). Let e be a road section, and appending it to the free trajectory point generates a restrained trajectory point. A restrained trajectory point is defined as a quadruple (x, y, t, e). The trajectory points mentioned here onward are restrained trajectory points.

For example, the coordinates of restrained trajectory points A and B are, respectively, denoted as (x_a, y_a, t_a, e_1) and (x_b, y_b, t_b, e_1), where e_1 represents the *Wangfujing* section; this implies that locations (x_a, y_a) and (x_b, y_b) are on road section e_1 at times t_a and t_b, respectively.

Definition 4 (restrained trajectory). A restrained trajectory, denoted as RT, represents a set of multiple restricted trajectory points:

$$\mathrm{RT} = \{\mathrm{Tid}, (x_1, y_1, t_1, e_1), (x_2, y_2, t_2, e_2), \ldots, (x_n, y_n, t_n, e_r)\}, \tag{1}$$

where Tid is the identification of a trajectory, time stamp t is arranged in the ascending order, implying that $t_s < t_{s+1}$, t $(1 \le s < n)$, n is the number of sampling, also called the length of the trajectory, and r is less than or equal to n. A road section can contain multiple locations, i.e., (x_1, y_1, t_1, e_1), (x_2, y_2, t_2, e_1), and (x_3, y_3, t_3, e_1), but a location belongs to only one road section; in addition, the road sections of adjacent locations are either identical or adjacent.

Furthermore, if e_1 and e_r values of two trajectories are the same, they are called neighbors. Consider a set of m trajectories $\mathrm{RTS} = \{\mathrm{RT}_1, \mathrm{RT}_2, \ldots, \mathrm{RT}_m\}$, where $\mathrm{RT}_i = \{\mathrm{i}$ $(x_1^i, y_1^i, t_1^i, e_1^i), \ldots, (x_q^i, y_q^i, t_q^i, e_r^i)\}$ represents the ith trajectory in RTS, $1 \le i \le n$.

Definition 5 (direction deflection angle). The direction deflection angle is defined as the degree of change at the position of a restrained trajectory point. Consider three consecutive trajectory points, denoted by $k(x_{i-1}, y_{i-1})$, $q(x_i, y_i)$, and $p(x_{i+1}, y_{i+1})$ (only the position of the direction deflection angle is considered in the calculation). Then, the direction deflection angle at trajectory point q is denoted as θ_q and is given by

$$\theta_q = \frac{\mathrm{distance}(p,q)^2 + \mathrm{distance}(q,k)^2 - \mathrm{distance}(p,k)^2}{2 * \mathrm{distance}(p,q)^* \mathrm{distance}(q.k)}. \tag{2}$$

Figures 1(a) and 1(b), respectively, show direction deflection angles less than 0 and greater than 0. Let A be the direction deflection angle of point q_1 on trajectory RT_1 and B be the direction deflection angle of point q_2 on trajectory RT_2. From equation (2), it can be seen that A is less than 0 and B is greater than or equal to 0.

Definition 6 (deme). Two trajectories RT_i and RT_j with their starting and ending points on the same road section are considered to be in a deme. Each deme includes two attributes, namely, the starting and ending road sections. According to the road section attributes of the starting and ending points, trajectories can be divided into various demes. The rth deme is denoted as D_r, with attributes $D_r.\mathrm{se} \in E$ and $D_r.\mathrm{de} \in E$, respectively.

Definition 7 (ATD-outlier). ATD-outlier includes three types of anomalies, namely, global router anomaly, local speed anomaly, and local shape anomaly. If a trajectory is an ATD-outlier, it can be at least one of the above anomalies.

4. ATD-Outlier Detection Algorithms

4.1. Framework Overview. In this section, we present an overview of our proposed framework. It contains two stages: the preprocessing stage and the anomaly detection stage, which contains three algorithms. As shown in Figure 2, in the trajectory data preprocessing stage, a map matching algorithm based on AntMapper [38] is used to match the trajectory points to the road sections. This method considers both local geometric/topological information and global similarity measures and uses an ant colony optimization algorithm, which mimics the pathfinding process of ants transporting food in nature. In addition, local heuristics and global fitness are used to search for the global optimal value of the model with high matching accuracy. The framework of the anomaly detection phase is described as follows:

Global router anomaly detection algorithm: the similarity between trajectories in the same deme is used as the input to the isolation forest (iForest) algorithm that trains a suitable model to determine global router anomaly trajectories.

Local speed anomaly detection algorithm: the instantaneous velocities of trajectory points are clustered by DBSCAN for each road section. A trajectory having a sufficient number of speed anomaly points will be marked as a local speed anomaly trajectory.

Local shape anomaly detection algorithm: the direction deflection angle of each trajectory point is calculated. The deflection angle of the trajectory point on the same road section is used as the input of the LOF algorithm to determine the abnormal trajectory of the lane change.

4.2. Global Router Anomaly Detection Algorithm. Currently, taxi charges for public are calculated on the basis of a standard mileage. In order to make extra profits, some taxi drivers take their passengers via long routes to their destinations in the urban road network, thereby fraudulently increasing mileage. However, traffic authorities cannot

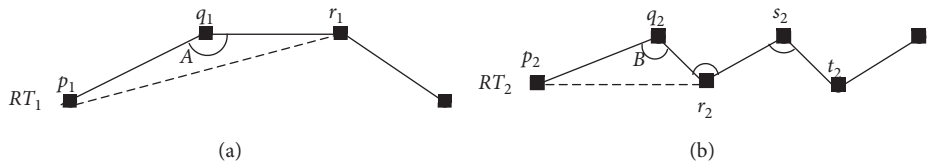

Figure 1: Examples of the direction deflection angle.

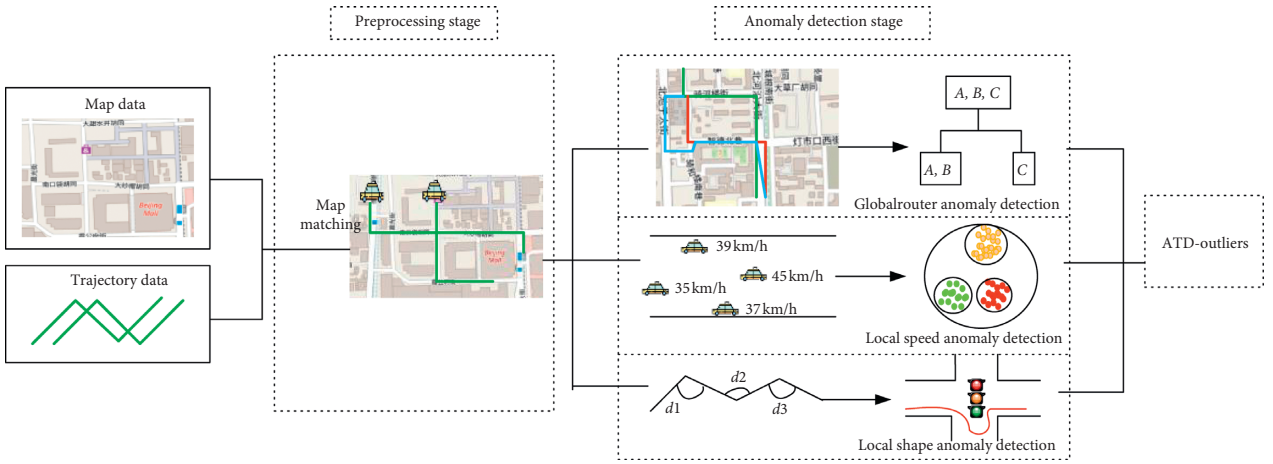

Figure 2: System framework diagram.

manually investigate and deal with such illegal behavior of taxi drivers. Therefore, a global router anomaly detection algorithm is proposed.

For example, as shown in Figure 3, a number of trajectories exist in a deme. The red line indicates an odd path that is different from the others (indicated by the yellow lines), for example, in terms of the length. This odd route is considered the global router anomaly trajectory as the driver has taken a very long route.

The distance between two trajectories determined by dynamic time warping (*DTW*) gives the similarity between the trajectories. The dynamic time regularity algorithm measures the similarity between two different time series. Using the distance function to determine the similarity between two trajectories that do not have a similar trip time is not feasible. However, if two trajectories have the same starting and ending points and a very similar time taken for the trips, they are comparable in terms of the distance function. Hereafter, in this paper, the reference to similarity between objects implies that the objects are in the same deme.

For example, let us consider comparison of a template trajectory sequence Q with an actual sampled trajectory sequence C; because of the different route patterns, both trajectories cannot be aligned. However, the first sampling value and the last sampling value of the two trajectories are taken such that they correspond pairwise to each other. Then, the process of calculating similarity is as follows.

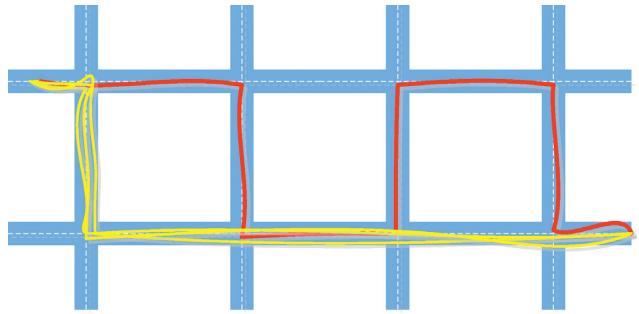

Figure 3: Example of the global router anomaly trajectory.

Step 1: we construct a $n \times m$ matrix, with elements $d(i, j) = \text{distance}(q_i, c_j)$. Without loss of generality, we utilize the Euclidean distance as the distance measure.

Step 2: the shortest path from $d(1, 1)$ to $d(m, n)$ is searched with dynamic programming. Because Q and C are both time series, there are only three directions to search.

Step 3: the similarity between trajectories Q and C is calculated by the shortest path from $d(1, 1)$ to $d(m, n)$.

Definition 8 (trajectory similarity). The similarity between two trajectories Q and C, denoted as SIM, is calculated as follows:

$$\mathrm{SIM}_{Q,C} = \gamma(i,j) = d\left(q_i, c_j\right) + \min\{\gamma(i-1,j-1), \gamma(i-1,j), y(i,j-1)\}. \tag{3}$$

In the equation, q_i and c_j, respectively, represent the ith point of Q and the jth point of C, and the similarity between Q and C is given by the value of $\gamma(n,m)$, where n and m, respectively, represent the lengths of Q and C, such that $1 \le i \le n$, and $1 \le j \le m$.

Definition 9 (deme similarity matrix). The deme similarity matrix, denoted as SM, is established for each deme. If $\mathrm{RT}_i, \mathrm{RT}_{i+1}, \ldots, \mathrm{RT}_k \in D_r$, $\forall \mathrm{RT}_i.e.s = \mathrm{RT}_j.e.s = D_r.\mathrm{se}$, $\mathrm{RT}_i.e.d = \mathrm{RT}_j.e.d = D_r.\mathrm{de}\,(i < j < k)$, the similarity matrix of the rth deme, denoted as SM_r, is calculated as follows:

$$\mathrm{SM}_r = \begin{bmatrix} \mathrm{SIM}_{1,1} & \cdots & \mathrm{SIM}_{1n} \\ \vdots & \ddots & \vdots \\ \mathrm{SIM}_{n1} & \cdots & \mathrm{SIM}_{mn} \end{bmatrix}, \tag{4}$$

where $\mathrm{SIM}_{i,j}$ is calculated by equation (3) and n is the number of trajectories in D_r.

When the number of trajectories is especially large in a deme, the dimension of this matrix is difficult to predict. Therefore, we set a constraint as follows:

$$\mathrm{maxtrix\_dimension}_r = \begin{cases} 10, & \text{if } \mathrm{len}\left(D_r\right) > 10, \\ \mathrm{len}\left(D_r\right), & 0 < \text{otherwise} \le 10, \end{cases} \tag{5}$$

where $\mathrm{maxtrix\_dimension}_r$ represents the dimension of the similarity matrix of D_r. When the number of trajectories in D_r is greater than 10, the dimension is set to 10. Otherwise, it is set equal to the number of trajectories. Another reason for setting the limitation is that it is not reasonable to use the attributes of trajectories with high similarity for anomaly detection. Consequently, we sort the similarity of each trajectory in the ascending order and select the attributes of the lowest ten similarities and use them as the input to the iForest algorithm. Theoretically, the length of the trajectory marked as abnormal should be longer than the length of the normal trajectories.

The iForest algorithm is an unsupervised anomaly detection method suitable for continuous data. It was first proposed by Professor Zhihua Zhou of Nanjing University in 2008 [39], and an improved version was proposed in 2012 [40]. Different from the other anomaly detection algorithms, which portray the degree of dissimilarity between samples is through distance, density, and other indicators, the iForest algorithm detects outliers by isolating sample points. Specifically, the algorithm isolates a sample using a binary search tree structure called the isolation tree or, in short, iTree. Because the number of outliers is small and alienated from most samples, the outliers are isolated earlier, that is, the outliers are close to the root node of iTree, whereas the normal samples are placed far from the root node. In addition, compared to traditional algorithms such as LOF and K-means, the iForest algorithm is more robust to high-dimensional data. Therefore, it is suitable for detour trajectories, which have ten dimensions. The specific Algorithm is as follows.

The time complexity of Algorithm 1 depends on the following aspects: (a) the time for calculating the *SM* matrix, whose time complexity is $(d \times s \times m)$, where d is the size of a deme and m and s are the length of trajectories, respectively; (b) the time of the iForest algorithm, whose time complexity is $o(n)$. To be precise, n is the largest size of a deme; therefore, the total time complexity is $o(\mathrm{len} \times s \times m \times n)$, where *len* is the size of a deme. The spatial complexity is $o(n^2)$, which is mainly due to storing of the *SM* matrix. The parameters of the iForest algorithm are the same as those used in the literature [37], so they are not listed.

4.3. Local Speed Anomaly Detection Algorithm. The local speed refers to the instantaneous speed of each trajectory point. Owing to the road section attribute, trajectory points are also classified. Then, the instantaneous velocity of trajectory points of each road section is clustered using the DBSCAN algorithm.

DBSCAN is a classical density-based clustering algorithm, having the following main characteristics: (1) the number of clusters does not need to be specified in advance when clustering and (2) the number of clusters is uncertain.

The correlative concept definitions of this section are presented as follows.

Definition 10 (instantaneous velocity of the trajectory point). Consider two consecutive trajectory points, denoted as $p(x_i, y_i, t_i)$ and $q(x_{i+1}, y_{i+1}, t_{i+1})$; the instantaneous velocity of point q is obtained as follows:

$$\Delta v_q = \frac{\mathrm{distance}\,(p,q)}{t_{i+1} - t_i}. \tag{6}$$

Definition 11 (core point). The core point indicates a point within a radius Eps that contains more than ε points (where ε is the minimum number of points to form a dense region). A point that contains less than ε points within a radius Eps falls in the neighborhood of the core point, which is also called the boundary point. Moreover, the noise point is neither a core nor a boundary point.

As shown in Figure 4 if ε is set to 3, according to Definition 11, the red points are core points because there are three points in the red circles with a radius Eps. The radius of all circles is Eps. There are two points in the blue circle centered on point B, so B is not the core point, but it falls within the red circle, so point B is the boundary point. Because the number of points in the green circle with C as the center is less than 3, C is not the core point, and because C does not fall within the red circle, it is not the boundary point, so C is the noise point.

Definition 12 (Eps-neighborhood). The neighborhood within a given object radius Eps is called Eps-neighborhood of the object. We denote the set of points within a radius Eps of point p as $N\_\mathrm{Eps}(p)$:

Input: RTS (a dataset of restrained trajectories)
Output: *OUT* (a dataset of the number of abnormal trajectories)
(1) *MD* ⟵ Extract the road section identifier value of the starting and ending points of all trajectories in RTS;
(2) *MD* ⟵ Delete duplicate pairs of road section;
(3) *deme* ⟵ Classify RT by *MD*;
(4) *len* ⟵ |*deme*|;
(5) for *i* ⟵ 1 to len do:
(6) Initialize the matrix *SM*;
(7) Calculate *SM* of the trajectories included in *deme*[i] using (4);
(8) *OUT*.append(iForest(*SM*));
(9) endfor
(10) return *OUT*;

Algorithm 1: *GRAD*: global router anomaly detection algorithm.

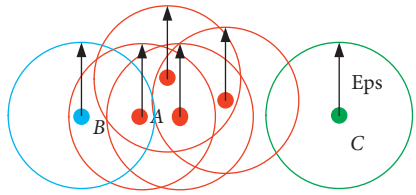

Figure 4: Example of core point, boundary point, and noise point.

$$N_{\text{Eps}}(p) = \{q | qt \epsilon n\, qD, distance(p, q) < \text{Eps}\}. \quad (7)$$

D is a given object set.

Definition 13 (density connection). Given an object set D if p is in the Eps-neighborhood of q and q is a core object, then object p is defined as directly density reachable. If there is an object chain $p_1, p_2, \ldots, p_n, p_1 = q, p_n = p, p_i \in D\,(1 \le i \le n)$, where p_{i+1} is the direct density from p_i about Eps and ε, then object p is defined as density reachable from object q with Eps and ε. If there is an object $O \in D$ and objects p and q are density reachable from about Eps and ε, then objects p and q are defined as density connection about Eps and ε.

Principle of judging abnormal points:

Step 1: clusters are created by checking Eps-neighborhood of each point in the dataset; if the Eps-neighborhood of point P contains more than points, a cluster is created with P as the core object.

Step 2: then, the aggregation of iterations from these core objects gives directly density reachable objects; this process involves merging of some density reachable clusters.

Step 3: when no new points are added to any cluster, the clustering process ends. Points that are not classified into any class are suspected anomaly points.

Step 4: each trajectory is checked for whether consecutive points are marked in it. If so, such trajectories are outliers.

Speed of urban taxis is subject to traffic laws in various road sections, but it is difficult to accurately obtain the speed limit of each road section. However, clustering the instantaneous speed of moving objects on various road sections is a feasible solution. If the instantaneous speed of a moving object differs greatly from those of other moving objects, that object may be regarded as an abnormal one. The detection result can be applied to detection of over speeding in real life. The specific algorithm is as follows.

Line 6 of Algorithm 2 clusters the instantaneous speed of trajectory points for each road section and records the location of noise points in *CLUSTER*. The time complexity of Algorithm 2 depends on the following aspects: (a) the time for clustering, whose time complexity is $o(n \times \log n)$ in low dimensions, where n is the number of trajectory points; (b) the time of checking trajectories, whose time complexity is $o(n)$. Therefore, the total time complexity of Algorithm 2 is $o(n^2 \times \log n)$. The spatial complexity is $o(n^2)$, which is mainly due to storing of *SPEED* and *CLUSTER* matrixes.

4.4. Local Shape Anomaly Detection Algorithm. The implication of a local shape anomaly on the trajectory is an abrupt change in the direction of the trajectory. Such deviations are considered illegal if they occur at an intersection or successively.

The LOF algorithm is an unsupervised outlier detection method proposed by Berning et al. [31] in 2000. It is a representative algorithm among outlier detection methods based on density. The algorithm calculates an outlier factor LOF for each point in the dataset and determines whether it is an outlier factor by judging whether the LOF is close to 1. If the LOF is much greater than 1, it is considered an outlier factor, whereas if it is close to 1, it is a normal point. Herein, we only provide a brief introduction to the concept of the algorithm, see [31], for further details.

This study mainly uses the LOF algorithm to detect the anomaly of the deflection angle of the track point on the same road section. The direction deflection angle of a trajectory point is evaluated by equation (2). The specific algorithm is shown as follows.

The time complexity of Algorithm 3 depends on the following aspects: (a) the time of calculating the direction deflection angle of trajectory points; (b) the time required to

```
Input: E (a dataset of road sections), RTS (a dataset of restrained trajectories)
Output: OUT (a dataset of number of trajectories with speed anomaly)
(1) Initialize the two matrixes: SPEED and CLUSTER;
(2) Initialize the array OUT;
(3) SPEED ⟵ calculate instantaneous velocity of each trajectory point;
(4) p ⟵ |E|;
(5) for i ⟵ 1 to p do:
(6)     Cluster instantaneous velocity of RTS on the i-th road section;
(7)     Delete the trajectory that only have an exception in a period of time;
(8) endfor
(9) Store the restrained trajectory number with abnormal speed at OUT;
(10) return OUT;
```

Algorithm 2: *LADA* local anomaly detection algorithm.

```
Input: RTS (a dataset of restrained trajectories)
Output: OUT (a dataset of number of abnormal trajectories)
(1) Initialize the array OUT;
(2) p ⟵ |E|;
(3) for i ⟵ 1 to p do:
(4)     Initialize the ANGLE matrix;
(5)     Initialize the array LOF;
(6)     ANGLE ⟵ calculate the direction deflection angle of each trajectory point on the road section i
(7)     LOF ⟵ local outlier factor()/* The direction deflection angle of all track points on the road section i is used as the input of
        the local outlier factor function, and it is judged whether a trajectory point is abnormal according to the set threshold; the abnormal
        point is −1*/;
(8)     If there are more than two abnormal points near the determined abnormal point, the track where the abnormal point is located
        is considered abnormal;
(9) endfor
(10) Store the restrained trajectory number with local shape anomaly at OUT;
(11) return OUT;
```

Algorithm 3: LSAD local shape anomaly detection algorithm.

execute the LOF algorithm. The LOF algorithm must calculate the distance between the two data points, resulting in the time complexity of the entire algorithm, $o(n^2)$, where n is the total number of all track points on road section i. The time complexity of calculating the direction deflection angle of trajectory points is $o(n)$; therefore, the total time complexity of local shape anomaly detection is $o(n^2 * p)$. The spatial complexity is $o(n^2)$, which is mainly contributed by the storing of the array *ANGLE*.

In this section, we introduced three different anomaly detection algorithms to detect three illegal behaviors of taxi drivers and established the algorithms by analyzing the characteristics of taxi trajectories, such as the characteristics of road segments and local characteristics of trajectories. Combining the three anomaly detection algorithms will save considerable labor cost and ensure safety and convenience of human travel.

5. Experiments

Experiments were conducted using Python3.7, with the software and hardware environment being Intel Core i5 @ 2.30 GHz quad-core CPU, 16G memory, and Windows 10 operating system.

The dataset is described in Section 5.1. We compare the three detection algorithms with a previous algorithm for trajectory neighbor (TN)-outlier [37].

5.1. Dataset Selection. The dataset contains the GPS trajectory data of 10357 taxis in Beijing for a period from February 2 to February 8, 2008. A total of approximately 15 million points are present in the dataset. The total distance of the trajectory reaches 9 million kilometers. Figure 5 plots the distribution of time interval and distance interval between two consecutive points. The average sampling interval between two points is approximately 177 s, and the average distance between two points is approximately 623 m. The figure indicates that the sampling frequency has approximately 50% of the trajectory points within three minutes. Figure 6 shows the density distribution of the GPS points in the dataset. The dataset provides all data for each taxi. Therefore, we considered that the user trajectory changes when the

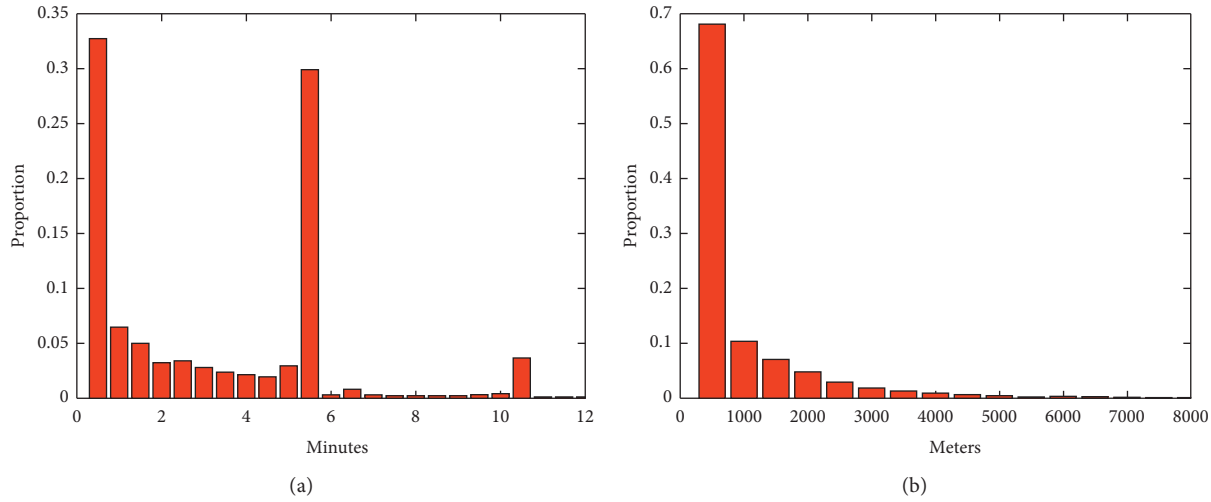

Figure 5: Histograms of time and distance intervals between two consecutive points: (a) time interval and (b) distance interval.

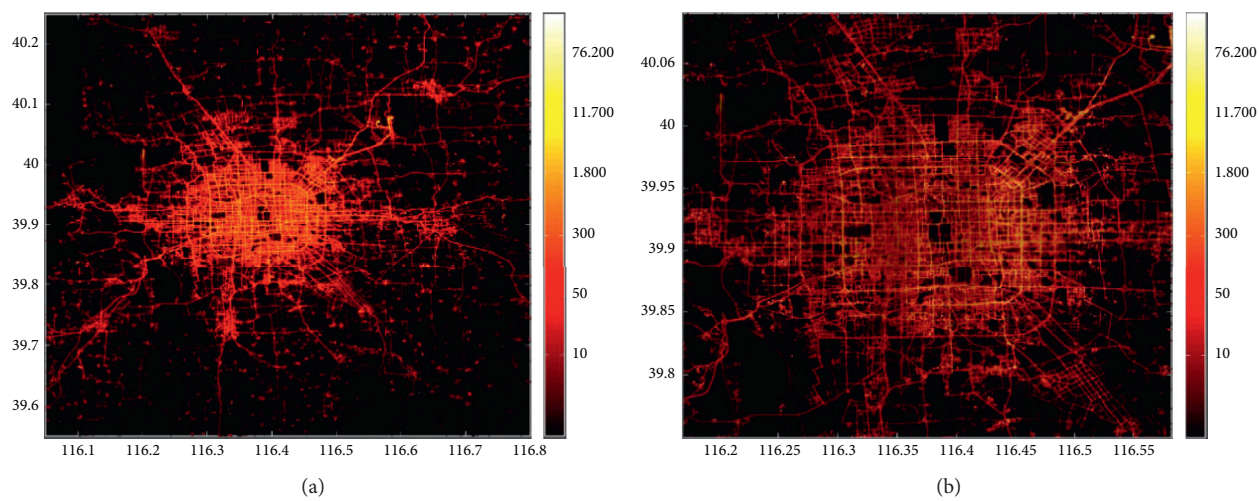

Figure 6: Distribution of GPS points, where the color gradient indicates the density of the points: (a) data overview in Beijing and (b) data within the 5th ring road of Beijing.

sampling interval is greater than 10 min. Moreover, the experimental data with the taxi trajectory length less than 10 are disqualified from the dataset.

To evaluate the precision of trajectory outlier detection, we selected the GPS trajectory of 5000 taxis, of which 7000 trajectories were screened out. We used the trajectory anomaly detection algorithm presented in the literature [25, 26, 31] to mark the 7000 trajectories. The trajectories marked as normal by these algorithms are regarded as true normal trajectories, whereas those marked as abnormal are regarded as true abnormal trajectories. In this manner, 3192 trajectories were marked as normal and 1186 trajectories were marked as abnormal. Due to the particularity of local shape anomaly detection and because the dataset used in this study has a low sampling frequency, we used cubic spline interpolation to interpolate the original trajectory.

5.2. Parameter Setting. In Algorithm 1, we selected the lowest ten similarities as trajectory attributes to detect anomaly. An enormous trajectory that did not have a neighbor was deemed abnormal. For a trajectory having a number of neighboring trajectories, we selected the minimum number of trajectory neighbors in the dataset as the number of attributes. However, the number of attributes must be greater than or equal to 2 and less than or equal to 10 because if the data dimension is extremely large, prediction by iForest may not be suitable.

The parameters Eps and ε vary for different applications and datasets. The first road data derived from OpenStreetMap show that a latitude and longitude of 0.00046 corresponds to an actual distance of 39.3 m. In Algorithm 2, we used a taxi speed of 40 mph as the standard speed, which is 64.2 km/h, given that 1 mile is 1605 m. We set ε between 1

and 10 m/s and converted it to latitude and longitude, that is, the Eps range was 1.17048×10^{-5} to 1.17048×10^{-4}. To facilitate the calculation, we expanded the data by 10,000 times such that Eps was normalized between 0 and 1. Then, Eps was set between 1 and 6.

5.3. Experimental Results. Some results of our methods were compared with the results of the TN-outlier detection algorithm, which is one of the most popular trajectory outlier detection algorithms. The trajectory data in this study was required to be preprocessed for map matching. We used the AntMapper algorithm [38] to match the trajectory points to the road section and then classified the starting and ending pairs. This algorithm uses an ant colony optimization algorithm that mimics the pathfinding process of ants transporting food in nature. It uses local heuristics and global fitness to search for the global optimal value of the model. For a 5-min sampling frequency, this algorithm could achieve a matching accuracy of 93.97%.

5.4. Visual Display of the Global Router Anomaly Detection Result. To illustrate that each trajectory can find corresponding neighbors under a large data volume, we randomly selected trajectories of three taxis and categorized them. When two sampling points of a taxi trajectory exceeded 10 min, the other trajectory was considered to have begun. The different colored lines in Figure 7 represent different trajectories, and each submap represents a taxi journey from February 2 to February 8, 2008. Clearly, most of the trajectories are concentrated in certain places where the passengers are transported back and forth; therefore, it is reasonable to classify the trajectories according to the departure and destination locations.

In Algorithm 1, for learning using the iForest algorithm, the parameters used in the literature [39] were adopted. Figures 8(a) and 8(b), respectively, illustrate the detour detection results of the global router anomaly detection algorithm and TN-outlier algorithm on the real taxi trajectory dataset.

Owing to the large number of demes in this dataset, we selected some demes to show the detection results in Figure 8. In the figure, the red lines indicate the trajectories that are detour or without neighbors and, hence, anomalies. The blue lines indicate normal trajectories. From the figure, we can observe that the abnormal trajectory is longer than the normal and that, in the middle of the trip, the abnormal trajectory increases the distance to the destination by taking some other road sections than normal.

The number of abnormal trajectories shown in Figure 8(b) is less than that in Figure 8(a). TN-outlier detection, as shown in Figure 8(b), can also detect trajectories without neighbors but not detour trajectories. This is because the TN-outlier detection algorithm does not account for the fact that a taxi increases the distance to the destination by detour, but only analyzes the shape characteristics of the trajectory and trajectory point neighbors. The global router anomaly corresponds to the long detour behavior of taxi drivers for gaining a higher profit than the profit without a detour. Of course, the long detour may be chosen by the taxi driver due to traffic jams or road repairs. In case of force majeure, most drivers may choose a longer trip, so the taxi driver's neighbors can be found in the route; therefore, this situation is not a global router anomaly. The global router anomaly detection can be used to track the itinerary of taxis or cars hired through online booking as a measure to protect the interests and safety of passengers.

5.5. Local Speed Anomaly Detection. The abnormal detection of speed cannot be achieved in the trajectory. Furthermore, a restrained trajectory was marked with a road section label, while the velocity of each trajectory was clustered to determine the abnormal speed. In order to illustrate the different detection results with different values of Eps and ε, we set Eps between 0 and 1 and ε between 1 and 6. Figure 9 shows the average number of trajectories with abnormal speed on each road section.

Algorithm 2 was used to detect trajectory outliers by DBSCAN to cluster the instantaneous velocity of the trajectory points. Because of the large number of clusters in this dataset, we selected clustering results of three road sections between 13 : 00 and 14 : 00 on February 2, 2008, as shown in Figure 10. The blue points represent the normal and the black points represent exceptions. After the completion of the clustering, we checked each trajectory for whether it contained consecutive points that were marked. Such trajectories, if any, were outliers. The point where the speed is abnormal is not distinguishable by observing the trajectory; hence, we do not show the speed anomaly detection results. Local speed anomaly can be used for over speed detection without video surveillance. Installing video surveillance in every corner of the city requires considerable manpower, financial resources, and regular maintenance of the equipment. However, the speed detection of vehicles is crucial because numerous accidents of individual or multiple vehicles occur due to over speeding every year. When the instantaneous speed of a vehicle at multiple consecutive moments is substantially different from the speed of other vehicles in the same lane, it is regarded as a local speed abnormality. In the final calculation of the precision and recall rates, we add the results of speed detection.

5.6. Local Shape Anomaly Detection. Algorithm 3 provides local shape anomaly detection. Based on equation (2), we calculated the directional deflection angle for the real dataset. Then, we used the direction deflection angle of the track point on each road segment as the input of the LOF algorithm to detect the trajectory points with an abnormal deflection angle. Furthermore, trajectories that contain two or more such points were marked as abnormal. In the LOF algorithm, we assigned different values of k, d, and f to compare the total precision, and the final parameters were set to $k = 5$, $d = 0.5$, and $f = 0.15$.

Figure 11(a) shows the detection result of trajectory outliers based on the LSAD algorithm; the outliers are indicated by red lines, while normal trajectories are indicated by blue lines. In the left-middle of Figure 11(a), several trajectories with a zigzag shape are marked as abnormal, but

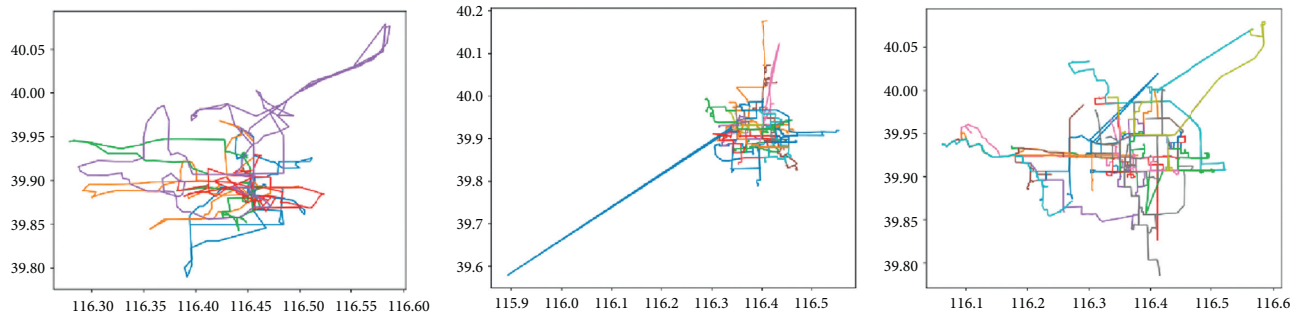

Figure 7: Trajectory slice: (a) taxi 1, (b) taxi 2, and (c) taxi 3.

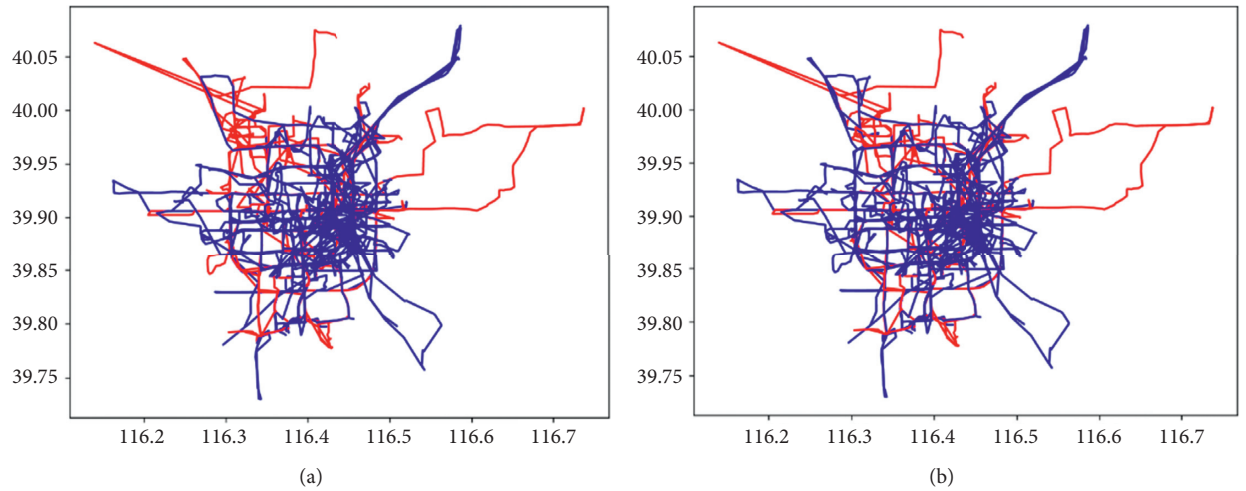

Figure 8: Detour detection by (a) GRAD (global router anomaly detection algorithm) and (b) TN-outlier detection algorithm.

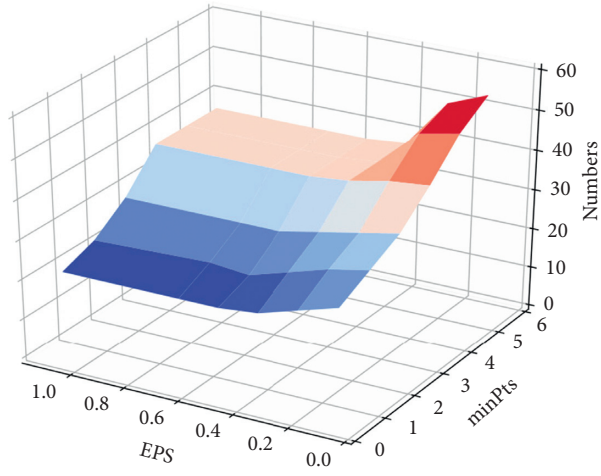

Figure 9: Average number of trajectories with abnormal speed.

in Figure 11(b) they are normal. These trajectories are determined by the LSAD algorithm as abnormal local shape. The LSAD algorithm combined with the road network analyzes whether the position of the point with a large degree of continuous curvature is at the intersection. The TN-outlier algorithm does not consider the feature. The road network data are too large to be clearly visible even after expanding the map, so the map is not displayed. Local shape anomaly may be caused by drunk driving or sudden sharp turns. Although the traffic inspection department considers

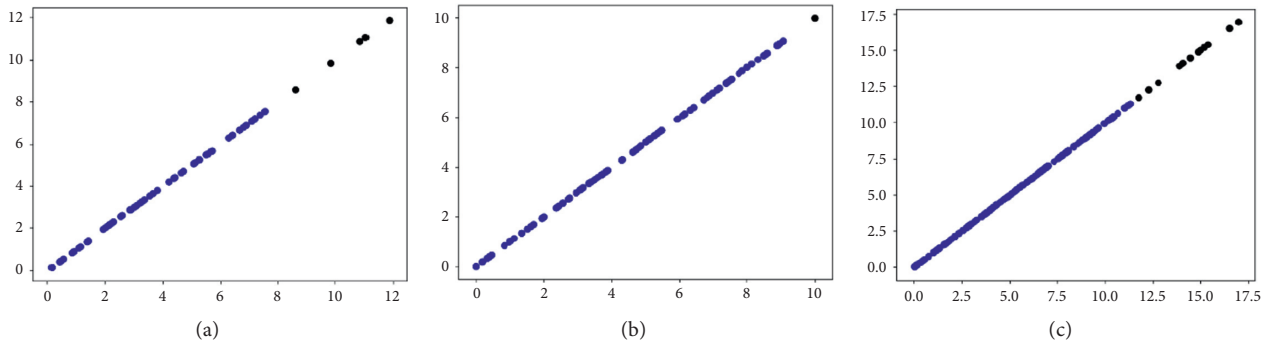

Figure 10: Speed cluster results: (a) road section 1, (b) road section 2, and (c) road section 3.

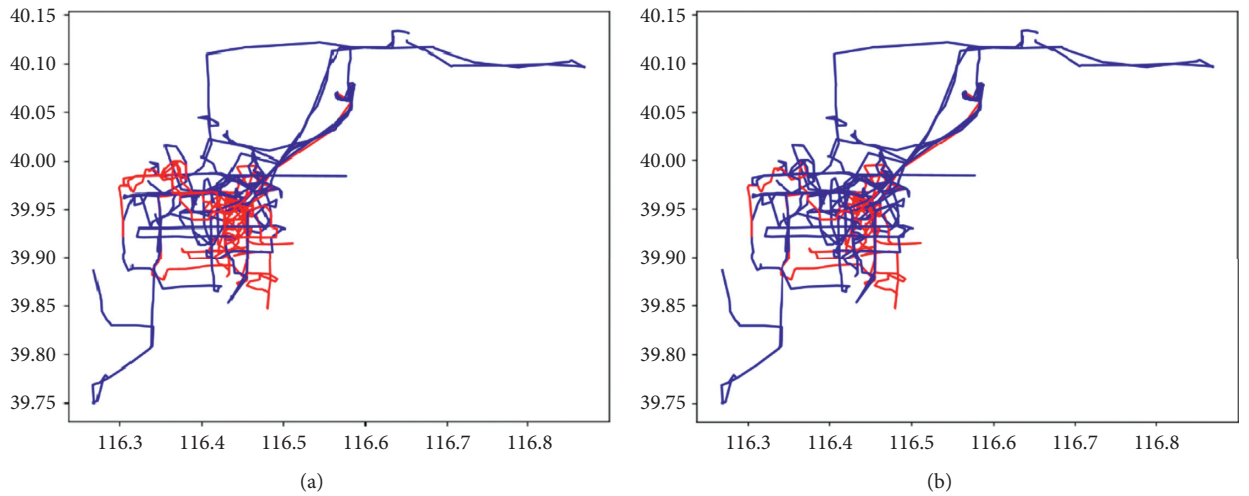

Figure 11: Deflection angle anomaly detection. (a) LSAD and (b) TN-outlier.

drunk driving a serious problem, it has always been tested manually, which requires considerable manpower and time. Local shape anomaly detection can be combined with trajectory semantics to determine whether the vehicle driver is drunk driving. If the starting point of the local abnormal trajectory is in a certain hotel, the driver is very likely to be drunk driving. The possibilities for local abnormalities are numerous, so we did not classify them specifically. If the local shape anomaly detection algorithm is applied to the traffic supervision system and combined with trajectory semantics, it can be further classified in detail.

5.7. Accuracy and Recall Rate of Abnormal Trajectory Detection. Next, we used precision and recall to measure the performance of abnormal trajectory detection. When calculating the precision and recall rate, the classification of abnormalities is not considered, but only whether the trajectory is abnormal is considered. Precision and recall are defined as follows:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$
$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{8}$$

where TP represents the number of detected abnormal trajectories, TN represents the number of detected normal trajectories, FP indicates the number of normal trajectories that are falsely detected as abnormal, and FN represents the number of abnormal trajectories that are falsely detected as normal trajectories.

Figures 12(a) and 12(b) show the precision and recall rate of the TN-Outlier detection algorithm. Although the recall is nearly 100%, the abnormal detection precision of the TN-outlier detection algorithm for taxi trajectories is not ideal.

Figure 13 shows the precision and recall rate of ATD-outlier. The number of abnormal trajectories in the precision and recall rate calculation is obtained by the union of the results of the three algorithms. The x-axis and y-axis of Figure 13(a), respectively, represent Eps and ε, and the z-axis represents the precision. In Figure 13(b), the z-axis represents the recall rate.

Although its recall rate is comparable to our ATD-outlier detection algorithm (Figure 13(b)), the TN-outlier algorithm considers a trajectory of a taxi as an outlier only if the taxi always moves alone. Moreover, the behavior of a taxi driver will be considered abnormal only if the driver always moves to regions that other taxi drivers hardly visit. Therefore, the TN-outlier detection algorithm frequently misclassifies trajectories of taxis.

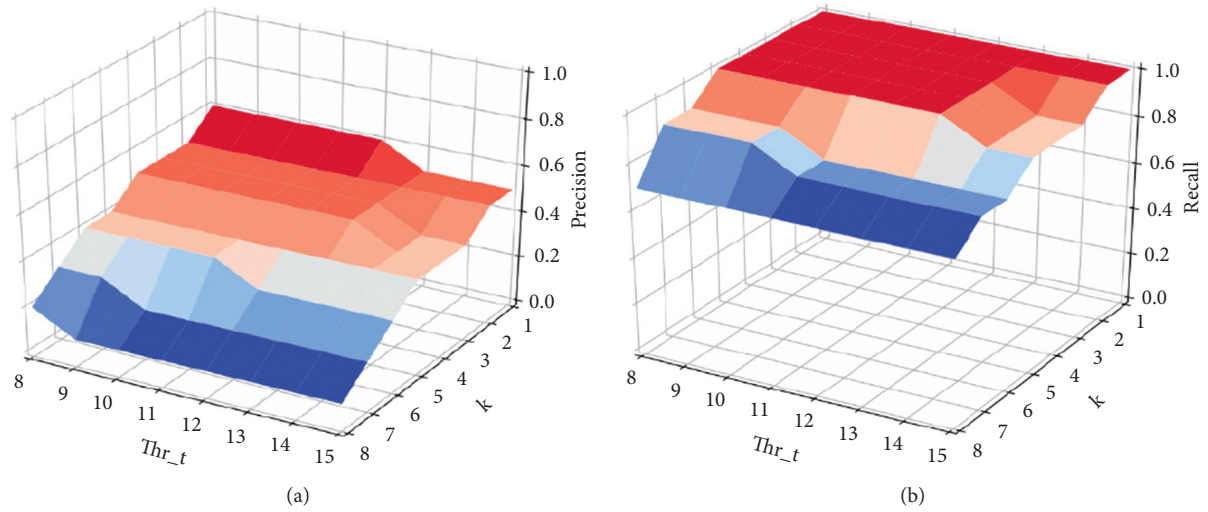

Figure 12: (a) Precision and (b) recall of the TN-outlier detection algorithm.

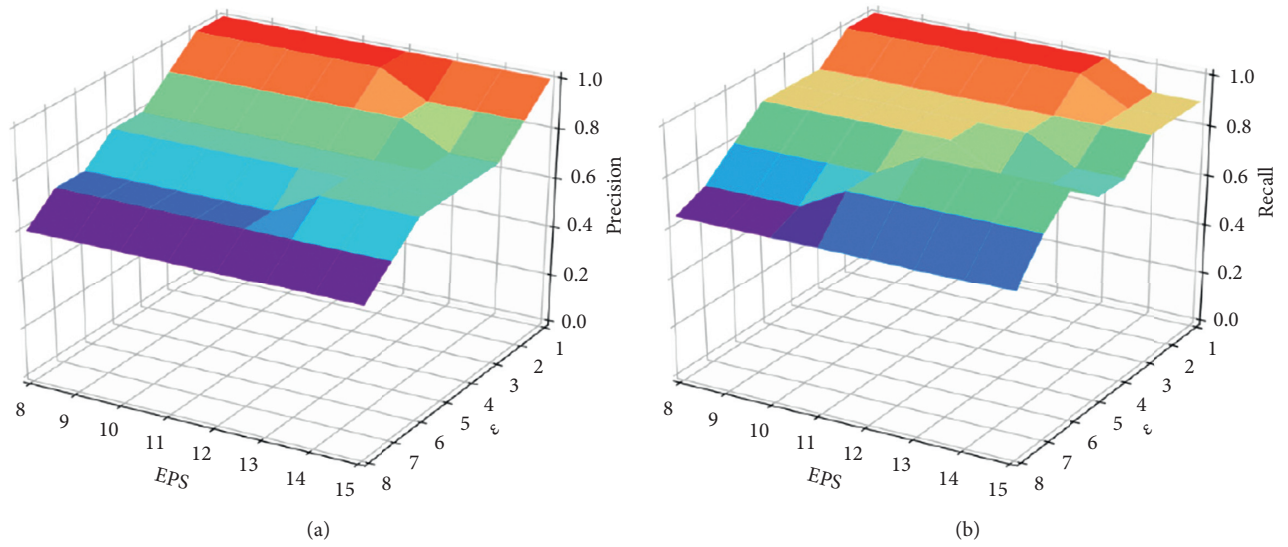

Figure 13: (a) Precision (b) recall of the ATD-outlier detection algorithm.

6. Conclusions

This study mainly focused on different anomalous features of trajectories and road network environment and proposed three corresponding detection methods. (1) Global router anomaly detection algorithm: according to the road section attributes of starting and destination points, the trajectories were first classified; then, abnormal trajectories in each deme were detected. (2) Local speed anomaly detection algorithm: the instantaneous speed of each trajectory was calculated; then, clustering algorithm was used to determine trajectories with abnormal speed. (3) Local shape anomaly detection algorithm: the trajectories with an abnormal deflection direction were determined on the basis of the direction deflection angle of trajectory points. Our framework contributes to city monitoring by analyzing big trajectory data under the mobile networks. Experiments to verify the algorithms were conducted using the Beijing taxi trajectory dataset of 2008. The results indicate that the proposed algorithms are better than an existing method tested for comparison. In general, the proposed methods can be applied in the construction of smart cities. The algorithm in this study roughly divides the abnormal trajectories into three categories according to the abnormal behavior of users. However, in actual situations, the classification of abnormal trajectories is complicated, and there are more than the three categories. In future work, we will perform further detailed anomaly classification for each type of anomaly and integrate time attributes and semantics, analyze road traffic, and provide personalized route recommendations because research based on real-time traffic of road sections is more meaningful.

Data Availability

The data used to support the findings of the study are available at https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61972439, 61672039, and 61702010). and the Key Program in the Youth Elite Support Plan in Universities of Anhui Province (gxyqZD2020004).

References

[1] Y. Huang, B. Li, Z. Liu et al., "ThinORAM: towards practical oblivious data access in fog computing environment," *IEEE Transactions on Services Computing*, vol. 13, no. 4, p. 602, 2020.

[2] W. Luo, H. Tan, L. Chen, and L. M. Ni, "Finding time period-based most frequent path in big trajectory data," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, June 2013.

[3] Z. Liu, B. Li, Y. Huang, J. Li, Y. Xiang, and W. Pedrycz, "NewMCOS: towards a practical multi-cloud oblivious storage scheme," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 4, pp. 714–727, 2019.

[4] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, August 2014.

[5] A. Hofleitner, R. Herring, P. Abbeel, and A. Bayen, "Learning the dynamics of arterial traffic from probe data using a dynamic Bayesian network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1679–1693, 2012.

[6] S. Liu, L. M. Ni, and R. Krishnan, "Fraud detection from taxis' driving behaviors," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 1, pp. 464–472, 2013.

[7] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," in *Proceedings of the 13th International Conference on Ubiquitous Computing*, Beijing, China, September 2011.

[8] F. Meng, G. Yuan, S. Lv, Z. Wang, and S. Xia, "An overview on trajectory outlier detection," *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2437–2456, 2019.

[9] J. Li, Y. Huang, Y. Wei et al., "Searchable symmetric encryption with forward search privacy," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2019.

[10] Z. Liu, J. Li, S. Lv et al., "EncodeORE: reducing leakage and preserving practicality in order-revealing encryption," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.

[11] Y. Djenouri, A. Belhadi, J. C.-W. Lin, D. Djenouri, and A. Cano, "A survey on urban traffic anomalies detection algorithms," *IEEE Access*, vol. 7, no. 7, pp. 12192–12205, 2019.

[12] J. Bian, D. Tian, Y. Tang, and D. Tao, "Trajectory data classification," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 4, pp. 1–34, 2019.

[13] T. Xiang and S. Gong, "Video behavior profiling for anomaly detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 893–908, 2008.

[14] A. Rehman and T. Saba, "Features extraction for soccer video semantic analysis: current achievements and remaining issues," *Artificial Intelligence Review*, vol. 41, no. 3, pp. 451–461, 2014.

[15] Y. Pritch, A. Rav-Acha, and S. Peleg, "Nonchronological video synopsis and indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1971–1984, 2008.

[16] S. Venugopalan, M. Rohrbach, J. Donahue, R. J. Mooney, T. Darrell, and K. Saenko, "Sequence to sequence—video to text," in *Proceedings of the International Conference on Computer Vision*, Santiago, Chile, December 2015.

[17] H. Y. T. Ngan, A. G. O. Yung, and A. G. Yeh, "Outlier detection in traffic data based on the Dirichlet process mixture model," *Iet Intelligent Transport Systems*, vol. 9, no. 7, pp. 773–781, 2015.

[18] R. J. Kingan and T. B. Westhuis, "Robust regression methods for traffic growth forecasting," *Transportation Research Record*, vol. 1957, no. 1, pp. 51–55, 2006.

[19] F. Chang, M. Li, P. Xu, H. Zhou, M. Haque, and H. Huang, "Injury severity of motorcycle riders involved in traffic crashes in Hunan, China: a mixed ordered logit approach," *International Journal of Environmental Research and Public Health*, vol. 13, no. 7, p. 714, 2016.

[20] R. Paleti, N. Eluru, and C. R. Bhat, "Examining the influence of aggressive driving behavior on driver injury severity in traffic crashes," *Accident Analysis & Prevention*, vol. 42, no. 6, pp. 1839–1854, 2010.

[21] J. Gou, W. Qiu, Z. Yi, Y. Xu, Q. Mao, and Y. Zhan, "A local mean representation-based K -nearest neighbor classifier," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 3, pp. 1–25, 2019.

[22] T. Kim, Y. Yue, S. Taylor, and I. Matthews, "A decision tree framework for spatiotemporal sequence prediction," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, August 2015.

[23] T. Lu, Z. Dunyao, Y. Lixin, and Z. Pan, "The traffic accident hotspot prediction: based on the logistic regression method," in *Proceedings of the 2015 International Conference on Transportation Information and Safety (ICTIS)*, Wuhan, China, June 2015.

[24] C. Piciarelli, C. Micheloni, and G. L. Foresti, "Trajectory-based anomalous event detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1544–1554, 2008.

[25] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: algorithms and applications," *The VLDB Journal*, vol. 8, no. 3-4, pp. 237–253, 2000.

[26] J.-G. Lee, J. Han, and X. Li, "Trajectory outlier detection: a partition-and-detect framework," in *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, Cancun, Mexico, April 2008.

[27] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: a partition-and-group framework," in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, Beijing, China, June 2007.

[28] I. San Román, I. Martín de Diego, C. Conde, and E. Cabello, "Outlier trajectory detection through a context-aware distance," *Pattern Analysis and Applications*, vol. 22, no. 3, pp. 831–839, 2019.

[29] Q. Yu, Y. Luo, C. Chen, and X. Wang, "Trajectory outlier detection approach based on common slices sub-sequence," *Applied Intelligence*, vol. 48, no. 9, pp. 2661–2680, 2018.

[30] Z. Zhu, D. Yao, J. Huang, H. Li, and J. Bi, "Sub-trajectory-and trajectory-neighbor-based outlier detection over trajectory streams," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Melbourne, Australia, June 2018.

[31] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Dallas, TX, USA, May 2000.

[32] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, Portland, OH, USA, August 1996.

[33] A. Kut and D. Birant, "Spatio-temporal outlier detection in large databases," *Journal of Computing and Information Technology*, vol. 14, no. 4, pp. 291–297, 2006.

[34] J.-G. Lee, J. Han, X. Li, and H. Gonzalez, "TraClass," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 1081–1094, 2008.

[35] G. G. Hazel, "Multivariate Gaussian MRF for multispectral scene segmentation and anomaly detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 3, pp. 1199–1211, 2000.

[36] S. A. Shaikh and H. Kitagawa, "Efficient distance-based outlier detection on uncertain datasets of Gaussian distribution," *World Wide Web*, vol. 17, no. 4, pp. 511–538, 2014.

[37] Y. Yu, L. Cao, E. A. Rundensteiner, and Q. Wang, "Detecting moving object outliers in massive-scale trajectory streams," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, August 2014.

[38] Y.-J. Gong, E. Chen, X. Zhang, L. M. Ni, and J. Zhang, "AntMapper: an ant colony-based map matching approach for trajectory-based applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 390–401, 2017.

[39] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, Pisa, Italy, December 2008.

[40] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 1, pp. 1–39, 2012.