Research Article

Grid-Based Anomaly Detection of Freight Vehicle Trajectory considering Local Temporal Window

Zixian Zhang1,2, Geqi Qi1,2,3, Avishai (Avi) Ceder1,4, Wei Guan1,2, Rongge Guo1,2, and Zhenlin Wei1,2

1Key Laboratory of Transport Industry of Big Data Application Technologies for Comprehensive Transport, Ministry of Transport, Beijing Jiaotong University, Beijing 100044, China
2School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China
3Beijing Research Center of Urban Traffic Information Sensing and Service Technologies, Beijing Jiaotong University, Beijing 100044, China
4Faculty of Civil and Environmental Engineering and the Transportation Research Institute, Technion-Israel Institute of Technology, Technion City, Haifa 32000, Israel

Correspondence should be addressed to Wei Guan; weig@bjtu.edu.cn

Received 18 April 2021; Revised 23 July 2021; Accepted 12 August 2021; Published 31 August 2021

Academic Editor: Dung-Ying Lin

Copyright © 2021 Zixian Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The security travel of freight vehicles is of high societal concern and is the key issue for urban managers to effectively supervise and assess the possible social security risks. With continuous improvements in motion-based technology, the trajectories of freight vehicles are readily available, whose unusual changes may indicate hidden urban risks. Moreover, the increasing high spatial and temporal resolution of trajectories provides the opportunity for the real-time recognition of the abnormal or risky vehicle motion. However, the existing researches mainly focus on the spatial anomaly detection, and there are few researches on the real-time temporal anomaly detection. In this paper, a grid-based algorithm, which combines the spatial and temporal anomaly detection, is proposed for tracing the risk of urban freight vehicles trajectory by considering local temporal window. The travel time probability distribution of vehicle historical trajectory is analyzed to meet the time complexity requirements of real-time anomaly calculation. The developed methodology is applied to a case study in Beijing to demonstrate its accuracy and effectiveness.

1. Introduction

With respect to urban security, logistics trajectories security plays an important role, which deserves a closer look. In particular, during major events such as the ceremony, concert, and games, real-time monitoring and anomaly detection of the logistics vehicles trajectories have attracted a large amount of security concern since the tragedy in Nice (86 dead and 458 injured) [1] and Barcelona (15 dead and 100 injured). As the vehicle trajectory data can be conveniently collected nowadays by digital sensors, such as various navigation systems, smart cellphones, and RFID devices, researchers can study the reliability of trajectories and the invoked transportation risk issues, i.e., driving behavior detection [2, 3], travel pattern recognition [4–6], and anomaly tracking in practice [4–10].

The unusual change of real-time trajectory may indicate the potential risk in spatial and temporal dimensions. Anomaly trajectory detection aims to find a trajectory that is significantly different from most of the historical trajectories with the same origin and destination. It is mainly reflected in the process of freight vehicles driving from the station to the distribution point. Due to the behavior habits of drivers and the periodicity of urban traffic, freight vehicles have similar driving paths with the same starting and ending points at the same time. In the field of anomaly detection of trajectory, many studies have been investigated from different perspectives [11–13], some of which are used to detect deceptive driving behavior in taxi drivers. However, these studies mainly focus on the spatial anomaly detection, and there are few related to the temporal anomaly detection. In terms of existing research methods, the research on space abnormal...
trajectory detection can be segmented into three main categories. The first category is based on distance and density [14, 15]. This category of method usually requires calculating metrics based on the entire historical database, which leads to high time complexity. The second category is based on architecture, mainly neural network architecture and reinforcement learning architecture [16, 17]. This type of method requires a large number of tags, and its accuracy is directly related to the quality of the data. The third category is pattern based [11, 13]. This type of method mainly studies the cooccurrence of trajectory points, so that abnormal trajectories with low probabilities can be detected in a preset window.

However, there are few related researches on real-time temporal anomaly discovery, which can help find the abnormal location of vehicles from the temporal perspective. This may be due to the following reasons: (1) urban transportation network flow is dynamic and complex, which leads to the different temporal security assessment threshold corresponding to different spatial locations, such that intersection parking and road section parking are different; (2) spatial anomaly can be judged by the isolated trajectory point, while temporal anomaly detection required a detection window cooperating multiple sequential trajectory point; (3) in real-time detection, selecting an appropriate detection window size is critical, and unreasonable selection may lead to completely wrong judgment. In extreme example, with only one trajectory point in the window, the travel time anomaly of a single trajectory point may be slight. However, with the expansion of time window, it is possible to exceed the threshold set by the detection window of historical trajectories.

In light of this, the window time of isolation based online anomalous trajectory detection (WT-iBOAT) algorithm is proposed in this paper to accomplish the real-time anomaly detection of abnormal logistics trajectories. The research can help supervise the driver of logistics vehicles to drive the vehicle safely and reliably and enable the real-time anomaly detection of freight vehicle trajectories by considering both spatial and temporal variations. The main contributions are as follows:

(i) First, we extract different constraints from not only spatial, but also temporal dimensions for comprehensively solving the motion-based anomaly detection problem

(ii) Second, a grid-based anomaly detection method is proposed from a new perspective of using the local temporal window, which is beneficial for increasing calculation accuracy and reducing the computation complexity of real-time detection

(iii) Third, this paper proposes a framework for detecting real-time risk trajectories of urban freight vehicles, and a real-world data set is utilized to verify its feasibility

The paper is organized as follows. In Section 2, the relevant work is investigated. In Section 3, related architectures and algorithms are proposed. In Section 4, the results of three trajectory datasets are analyzed using the algorithm proposed in this paper. In Section 5, the performance of the algorithm is analyzed. In Section 6, the adjustment mechanism of the algorithm is discussed. In Section 7, we summarize and discuss the whole paper.

2. Related Research

The trajectory data have been widely used in the popular route discovery and anomaly detection. In the field of popular route discovery, Zaiben et al. [18] proposed a coherence expanding algorithm and a maximum probability product algorithm to discover the most popular route. Jani et al. [19] used a fast map server, which could generate and visualize hot maps of hot routes online from massive motion trajectory data according to customers’ preferences. Luo et al. [20] found that suffix-optimal, length-insensitive, and bottleneck-free methods reflect common-sense notions and proposed efficient search algorithms to determine the most frequently used paths based on the time period. The semantic features mainly come from other moving objects with similar behaviors. Treerapot et al. [21] proposed a statistical approach to estimate truck activities and freight analytics from Global Positioning System (GPS) data of trucks and applied the method in estimating activities, activity patterns, commodity trip chains, and status of trips legs from the collected truck GPS data. Wei et al. [22] studied the characteristics of uncertain trajectories and proposed a route inference framework based on a collective knowledge algorithm to find the top-k popular routes. Outlier detection is important work in mining trajectories [23, 24]. Lee and Cho et al. [24] proposed a grid-based local outlier factor algorithm for outlier detection. This algorithm reduces the computation time significantly with acceptable trade-off errors. Knorr et al. [23] studied the notion of distance-based outliers to find outliers in multidimensional datasets. Discovering the popular route is always used for daily management and operation of urban traffic. Sometimes, it is helpful to find out the abnormal trajectories, but it lacks directness compared with anomaly detection and may result in misidentification of some regular trajectories (not most popular ones).

In the field of anomaly detection, Wang et al. [25] used improved Robust Principal Component Analysis (RPCA) to introduce the detection of abnormal passenger flow and Spatial Temporal-Density-Based Spatial Clustering of Application with Noise (ST-DBSCAN) algorithm to group the detected station-level anomalies. Suzuki et al. [26] applied the hidden Markov model (HMM) and k-means, and Bera et al. [7] used Bayesian learning techniques to detect abnormal pedestrian trajectories. Other popular fields related to GPS traces include the detection of social communities [27] and abnormal driver behavior [28]. Yin et al. [27] proposed a probabilistic model, which was used to infer users’ social communities by integrating their spatiotemporal data and semantic information. Ali and Ismail [28] investigated driver behavior through GPS trajectory data from the vehicle, mainly considering the features of speed limit, sudden speed changes, and constant changes in the
vehicle lateral position, and they used these features to detect abnormal driver behavior.

Many researchers have conducted related research on abnormal trajectory detection. In Song et al. [16], in order to solve the problem, i.e., the traditional anomaly trajectory detection method cannot adequately capture the sequence information of the trajectory, and focusing only on the given source and target may lead to data sparseness. They proposed an algorithm that used an Anomalous Trajectory Detection using Recurrent Neural Network (ATD-RNN) to represent the trajectory by learning a trajectory embedding and then found the risk trajectory. Liping et al. [15] put forward a trajectory segmentation strategy based on multimotion features and a similarity measurement method based on trajectory structures. With the proposed strategy and method, a distributed trajectory clustering algorithm is designed to improve the efficiency of the clustering algorithm. To tackle the problem of calculating distance and neighborhood density in the trajectory anomaly detection algorithm, the dataset was pruned by the trajectory clustering results, which improves the efficiency of the algorithm. Nonetheless, Song et al. [16] and Liping et al. [15] required artificial label of abnormal trajectory for improving effective of algorithm and could not perform real-time anomaly detection.

Wu et al. [17] proposed a probabilistic model-based vehicle outlier detection method, Driving Behavior-based Trajectory Outlier Detection (DB-TOD), which was used to model the driving behavior in a historical trajectory set. They designed outlier detection algorithms for both full and partial outliers, which solved the problem for providing early warnings of outliers only when the outliers were partially observed. Lee et al. [8] et al. aimed to address the problem that the detection algorithm for tracking outliers has limited ability in detecting the outliers’ tracks. An outlier detection algorithm, time-dependent popular route-based trajectory outlier detection (TPRO), was proposed. The algorithm was divided into two stages: partitioning and detection. For the first stage, a two-layer trajectory partition strategy has ensured the high quality and high efficiency of subtrajectory partitioning. In the second stage, the hybrid outlier detection method based on distance and density clustering improves the accuracy of the trajectory outlier detection algorithm. Daqing et al. [11] proposed an isolation-based anomalous trajectory (iBAT) method based on isolation. First, the original trajectory was mapped into the grid sequence, and the historical grid trajectory index was established. Then, the iForest algorithm was used to detect the isolated trajectory. However, this approach can only support full trajectories. Sun et al. [29] proposed the iBOAT algorithm based on trajectories, similar to the real-time detection method, which was an influential method. This method was developed from the iBAT algorithm. The main improvement of this method is the way that it compares with iBAT in view of the whole line offline, as it is improved by using the method of real-time contrast. It is more practical, because it can be a fast detection algorithm with high reliability. Yu et al. [30] defined anomalous behavior of moving objects according to the idea of neighbor-based trajectory outliers and proposed the minimal examination framework to effectively discover new abnormalities. Yingyi et al. [31] proposed a method of monitoring distance-based anomalies over trajectory streams. This method mainly used local clustering and piecewise vantage point trees with the corresponding pruning method to reduce the calculation time. These studies can make efficient detecting trajectory outlier while preserving the effectiveness. But these algorithms only study space anomaly detection and do not relate the temporal anomaly detection.

Jie et al. [12] used a historical trajectory dataset and commonly detected abnormal trajectories. Then, they modified the TPRO algorithm by dividing the steps into offline processing and online detection and adding appropriate modules to adapt the detection. This paper proposed an algorithm named trajectory outlier detection (TPRRO). TPRRO is a real-time outlier detection algorithm, and it constructs short for time-dependent transfer index (TTI) and short for time-dependent transfer graph (TTG) to conduct spatiotemporal anomaly detection. Although this paper proposed an algorithm for temporal anomaly detection, it did not analyze temporal anomaly detection in detail. At the same time, Jie et al. [12] used time-dependent transfer graph (TTG), which calculated the travel time from a vertex to another in road network, so this algorithm could not reflect immediate temporal anomaly information of trajectory.

In summary, we present a real-time temporal anomaly detection algorithm for logistic vehicle to prevent potential security incidents. Different from the previous study (in Table 1, we summarize contributions of previous study and our research), we focus on the real-time temporal anomaly detection method and design algorithmic framework for spatiotemporal anomaly detection of Urban Freight Vehicles. We further conduct analysis to our collecting data and explain the possible motivations and reasons of the anomalous behaviors.

3. Methods

In this section, the algorithmic framework of this paper is introduced, including data preprocessing (DP), offline processing (OFP), the data manager (DM), and online processing (OLP). The purpose of DP is to obtain the effective loading and unloading positions of raw GPS data. The goal of OFP is to obtain the available subtrajectories from the loading (origin) and unloading (destination) positions obtained from the DP and to mapping the subtrajectories to a preset grid. The goal of OLP is to real-time meshing test trajectory points. The DM is mainly used for extraction of spatiotemporal experience constrained from the historical trajectories and detect abnormal trajectories. Figure 1 shows the submodules within the modules.

3.1. Concept Relationship. This section first introduces the relationship among the concepts involved. For trajectory anomaly detection, the relationship between trajectory point and time/loading position/unloading positions is many-to-one, because many trajectory points are associated with the
same time/loading position/unloading position. For the same loading and unloading positions, the trajectories are different in different time periods. A loading position may be related to the trajectory of different roads, different times, and different unloading positions. In addition, trajectories crossing a specific road section may involve different loading positions; trajectories crossing a specific road section and different unloading positions may also involve similar relationships.

3.2. Meshing of the Spatiotemporal Trajectory Point. The movement of vehicles produces a lot of trajectory data. We need to divide the existing historical trajectory effectively before detecting the spatiotemporal anomaly of trajectory. The granularity of spatiotemporal trajectories is crucial when dividing trajectories. Different mesh sizes are suitable for different trajectory granularity problems. This is the reason why we use the grid method, which does not require us to know all the space-time coordinates, which avoid a lot of calculation. The trajectory point is mapped to the two-dimensional space of the grid cell (Figure 2). This is a two-dimensional Euclidean space defined by x-axis and y-axis. The center of each grid cell is used to represent all GPS points of the grid cell. If the grid cell is too large, the accuracy of anomaly detection may not be enough. However, the smaller grid cell may lead to much computation. Therefore, we must find an optimal grid size that can identify trajectories without unnecessary computation. Based on two-dimensional grid division, the trajectory query from specific loading position coordinates to specific unloading position coordinates becomes the trajectory query from one grid cell to another.

We define the minimum range of grid as \((\text{COLN}_{\text{min}}, \text{ROWN}_{\text{min}}, \text{COLN}_{\text{max}}, \text{ROWN}_{\text{max}})\) and the length range of a grid as \(k\). \(X\) and \(Y\) present the numbers of columns and rows of grid in the study area [32].

\[
X = \frac{\text{COLN}_{\text{max}} - \text{COLN}_{\text{min}}}{k} \quad (1)
\]

\[
Y = \frac{\text{ROWN}_{\text{max}} - \text{ROWN}_{\text{min}}}{k}
\]

3.3. Rough Selection of the Trajectory Set. Before anomaly detection of test trajectory, we first need to filter the historical trajectory, which has the same loading and unloading point as the test trajectory. We define this process as the query of historical trajectory. The related concepts are defined as follows:

Definition 1 (trajectory). Trajectories \(\text{tr} = \{P_1, P_2, \ldots, P_N\}\) are collected GPS records. Each GPS record \(P_i\) specifies the instantaneous time, position (latitude and longitude), and vehicle ID of the vehicle.

Definition 2 (subtrajectory). The subtrajectory reflects different driving preferences of different drivers for vehicles with the same transportation purpose in different periods. We use the effective parking time as the division factor of the subtrajectory. So, the subtrajectory is a part of the trajectory.

Definition 3 (scalable loading and unloading point of trajectories). In order to eliminate irrelevant trajectories, the loading and unloading points are used as spatial constraints to filter all trajectories. In order to achieve this goal, we first find loading points for all the subtrajectories and keep the subtrajectories. After that, we find the corresponding unloading point in kept subtrajectories. Compared with the common input method, this method has a huge advantage: a trajectory is used in anomaly detection when the middle section of this trajectory has the same loading and unloading points as the test trajectory, as shown in Figure 2.
3.4. Extraction of Spatiotemporal Constraint

Definition 4 (global spatial constraint). The different drivers have specific road choice preference in a certain period; this difference comes from drivers’ perception of the road traffic system. However, there are some uncertainties in this perception. For experienced drivers, the optional travel in a fixed time is regular and searchable. Specifically, the travel with the same loading and unloading positions generally follows optimal driving conditions. For example, the driver may require the minimum distance, time, and fuel consumption of the travel path. The example can be seen in Figure 3: O is the loading position, and D is the unloading position. The purple line represents the historical trajectory, and the red dotted line represents the test trajectory. The blue box represents the grid of online mapping. The yellow box indicates the coordinates of the grid.

In order to extract the spatial constraints of trajectory, we use part ideas of IBOAT algorithm. The specific methods are as follows. Suppose that there are three purple trajectories in Figure 3, \( T = \{ t_1, t_2, t_3 \} \), where \( t_1 = \{(1, 1), (5, 2), (7, 2), (10, 3), (13, 3), (17, 3), (21, 3), (24, 3), (27, 4)\} \), \( t_2 = \{(1, 1), (5, 2), (8, 2), (11, 3), (15, 3), (19, 3), (23, 3), (26, 3), (27, 4)\} \), \( t_3 = \{(1, 1), (5, 2), (7, 2), (11, 3), (13, 3), (19, 3), (21, 3), (25, 3), (27, 4)\} \), and a test trajectory \( t = \{(1, 1), (5, 2), (8, 2), (11, 3), (19, 3), (27, 3)\} \), then hasPath \( (T, t) = \{ t_2, t_3 \} \). So, hasPath is defined as a function that returns the set of trajectories from \( T \) (historical trajectory) that contains all of the points in \( t \) (test trajectory) in the correct order \[29\].

\[
\text{hasPath}(T, t) = \left\{ t' \in T \mid \begin{array}{l}
(1) 1 < i < n, t_i \in t' \\
(2) 1 < j < n, t' \cap t_j' \subseteq t'
\end{array} \right\}
\]

The support of the test trajectory is calculated as follows: in the grid coordinates (3, 2), the red trajectory and purple trajectory are in the same grid, and the support value of the test trajectory is 100%. The grid coordinates (9, 5) contain only the red trajectory. The support value of the test trajectory is 0% \[29\].
The support has a great influence not only on the spatial anomaly detection of trajectory, but also on the temporal anomaly detection. The modeling of real-time temporal anomaly depends on the spatial anomaly detection results. In other words, temporal anomaly detection is performed when the space is normal. Therefore, this condition is a strong constraint for temporally anomaly detection.

\[
support = \frac{|hasPath(T_i, t_i)|}{|T|}
\]  

(3)

**Definition 5** (local spatio constraint). The support considers the global spatial characteristics of trajectory but lacks the local spatial characteristics. The trajectory is composed of trajectory points that have different effects on the abnormality of the current trajectory point. In short, the closer the trajectory point is to the current test trajectory point, the greater the impact on the anomaly. On the contrary, the smaller the impact. In view of this characteristic, this paper adds two parameters, namely, \(t_2\) (length of the maximum continuous temporal anomaly...
window) and \( t_1 \) (length of the warning time window), to correct the results of the spatial anomaly. (In Section 6.2, we will explain in more detail.)

\[
R_t = 100 \times \frac{\min(N_h(t_1), N(t_2))}{N(t_2)}.
\]  

Equation (4) is the model for calculating \( R_t \) (the risk value of the trajectory). \( N_h(t_1) \) is the maximum number of abnormal GPS points of the test trajectory in \( t_1 \). \( N(t_2) \) is the number of all GPS points of the test trajectory in \( t_2 \). The longer the \( t_1 \) is, the lower the tolerance of the model, while the shorter the \( t_2 \) is, the higher the sensitivity.

**Definition 6** (global temporal xonstraint). Since the road network and traffic conditions are dynamic, the trajectory is also dynamic and depends on the traffic conditions. Therefore, the trajectories of the same loading and unloading positions may be completely different during peak hours and other times of the day. In terms of time, we divide trajectories according to the traffic conditions in different periods. Taking Beijing as an example, the morning and evening peaks are from 7:00 a.m. to 9:00 p.m. and from 5:00 p.m. to 8:00 p.m. Based on this, we divide the time into four periods: 7:00 am–9:00 am, 9:00 am–5:00 pm, 5:00 pm–8:00 pm, and 8:00 pm–7:00 am. We look for the same time as the test trajectory.

**Definition 7** (local temporal window constraint). The global temporally constraint classifies the trajectories from the overall time, but this is not sufficient for the real-time anomaly detection of the trajectory. Since the global temporally constraint span is too large to meet the real-time accuracy requirements of the trajectory, the extraction of the local temporally constraint with a smaller span will be the key to real-time anomaly detection.

A window time is used in equation (4), which inspired the idea of using sliding windows for temporal anomaly detection. The different window sizes provide different information, which is beneficial in the judgment of temporal anomalies. There are two types of sliding windows. One has a fixed time size, but the number of grids in the window is variable; the other has a fixed number of grids in the window, but the time is variable. This paper uses the second method, which is called fixed position window, because it is more intuitive to compare the travel time of the historical trajectory and the test trajectory in the same window.

**Definition 8** (Construct Empirical Local Temporal Window Constraints Based on Historical Information). According to the travel time preference obtained from all historical trajectories, travel temporal is divided into normal temporal and abnormal temporal. Based on the method of detecting outliers using box plots in anomaly detection, inspired by this, we use quantiles as a statistical reasoning tool to determine the abnormalities of travel temporal in window. In the historical trajectory, the trajectories that meet the conditions of Definitions 4 and 6 constitute the travel temporal frequency of all historical trajectories under a preset size window.

Let function \( c(t) \) be the travel temporal frequency for window, determined by counting the length of travel temporal in window that meets the above conditions in all trajectories.

\[
p(r) = P(c(t) = r), \quad (r > 0).
\]

Let \( M(r) \) be the number of windows that satisfy \( c(t) = r \). Let \( Y \) be the total number of windows. Then, the formula can be rewritten as

\[
p(r) = \frac{M(r)}{Y}.
\]

Let \( f(q) \) be the cumulative distribution function of \( r \).

\[
f(q) = \int p(r)dr, \quad (r \geq 0),
\]

Let \( Q_k(T_i, t_i) \) be the \( k \)-th quantile of the travel time probability distribution of the historical trajectories in the \( i \)-th window. We use a probabilistic method to model the distribution of the travel times of historical trajectories to detect temporal anomalies in the test trajectory.

\[
HT_i = Q_k(T_i, t_i), \frac{\text{hasPath}(T_i, t_i)}{|T|} > \theta.
\]

**Definition 9** (adjustment mechanism: local temporal window constraint extraction based on redundant information). In the anomaly detection problem, real data may occur, in which historical trajectory (HT) travel time is greater or less than the new test trajectory (NTW) travel time in the \( i \)-th window. In this case, the travel time in the \((i-1)\)-th window impacts the travel time in the \( i \)-th window. Therefore, we want to eliminate this error. We define this case as follows: the test trajectory in the window has redundant information. This effect is significant for temporal anomaly detection, which will lead to insufficient or overloaded historical information in window. These two situations directly lead to the failure of constraint extraction or not. Because the travel time of the \((i-1)\)-th window affects the travel time of the \( i \)-th window, therefore, we want to eliminate this error by using a linear equation and Definition 8.

Specifically, when HT is greater than NTW, if the HT value in the \( i \)-th window is not adjusted, the information in the \((i-1)\)-th window may be carried into the \( i \)-th window. This causes the HT value in the \( i \)-th window to be greater than the required HT. An adjustment coefficient \( (r_p) \) is needed in this situation.

HT is less than NTW in Figure 4. This is the opposite of the above situation, but it will also cause errors due to the introduction of redundant information. Therefore, the same needs to be adjusted.

The formula is as follows:

\[
HT_i = HT_i + r_p \sum_{j \in \Delta} \Delta t_{pj} + r_n \sum_{\Delta u} \Delta t_{nj}.
\]

The difference between the travel time of the historical trajectory and the travel time of the test trajectory is
Finally, we obtain the rule that the locations where \( \Delta t_i \) is followed by \( \text{timeStep5} \) in \( \text{TW}_8 \) in the historical trajectory, there are 30 seconds between \( \text{timeStep4} \) and \( \text{timeStep7} \). \( \text{timeStep4} \) means that time steps (5, 6, 7) are in the same grid. \( \text{k}\%_\text{here} \) is followed by \( \text{timeStep7} \) in \( \text{TW}_8 \) in the test trajectory, which is solved. \( \text{k}\%_\text{he} \) value of \( \text{HT} \) in \( \text{NTW} \). However, by comparing \( \text{HT} \) and \( \text{NTW} \), we discovered that this is a situation where we need to detect time anomalies. At this time, the estimation accuracy of the redundant value will affect the abnormal judgment of the trajectory. \( \text{NTW} \) is the travel time of the test trajectory; \( \text{HT} \) is the digital value of \( \text{HT} \) represents the time travel of the test trajectory, which is equal to the length of the window minus 1.

\[
\Delta t_i = \text{NTW}_i - \text{HT}_i. \tag{10}
\]

In equation (10), \( \text{HT}_i \) indicates the value of \( k \)-th quantile for travel time in \( i \)-th window.

**Figure 4: Example of trick.**

![Diagram](https://via.placeholder.com/150)

**Definition 10** (constructing empirical average speed ratio constraint). The change in average speed can express the change in travel time well. Therefore, the speed is referenced to help filter out the temporal anomaly in equation (6), where \( a \) is the speed ratio threshold, and \( WTR \) is the speed ratio of the \( i \)-th window. When \( WTR \) is greater than \( a \), the travel time of the \( i \)-th window is abnormal. The formula is as follows:

\[
WTR_i = \left| \frac{\text{speed}(\text{hasPath}(T_i.t_i)/|T| < \theta)) - \text{speed}(\text{NTW}_i)}{\text{speed}(\text{hasPath}(T_i.t_i)/|T| < \theta))} \right| > a. \tag{11}
\]

**Definition 11** (travel time of the test trajectory). \( \text{NTW}_i \) is the travel time of the test trajectory, which is equal to the length of the window minus 1.

\[
\text{NTW}_i = (\text{len}(\text{TW}_i) - 1). \tag{12}
\]

The above description is called window time of isolation based online anomalous trajectory detection (WT-iBOAT) algorithm, which adds an abnormal temporal detection method to BOAT [29]. Algorithm 1.

The pseudocode of WT-iBOAT algorithm is as follows:

**4. Results**

4.1. Data Description. To detect abnormal trajectories of urban freight vehicles, the historical trajectories provided by a logistics company in Beijing were used in this paper.

The data were collected from approximately 35 vehicles in Beijing in 2019 (September to October 2019). The main fields are the vehicle status, longitude, latitude, offset...
dimension, city code, working condition time, GPS time, accumulated mileage, positioning status, vehicle speed, vehicle ID, etc. It should be noted that these data are recorded every 10 seconds. We first need to segment the raw trajectory according to the business requirements of logistics. Therefore, loading and unloading points that are common locations for vehicles to load and unload goods are used as the segmentation points of the trajectory. Loading and unloading points are obtained by the effective parking time of the vehicle. The effective parking time depends on the experience of the staff. The regularity of vehicle trajectory can be found by the loading and unloading position obtained from the effective parking time. Clustering is a good choice to obtain this regularity. DBSCAN [33] is a density based algorithm, which assumes that the category can be determined by the compactness of the sample point distance.

DBSCAN is commonly used to find the loading and unloading positions. The parameters of the algorithm are minimum number of fields (MINPTS) = 60 and domain radius (EPS) = 150 meters. A total of 151 loading position categories and 151 unloading position categories are obtained. The clustering results based on the ArcGIS platform are shown in Figure 5. On the left is the loading position, and on the right is the unloading position. The loading and unloading positions in the central city are relatively concentrated, which meets the requirements of urban logistics, indicating that the clustering results reflect the common working area.

By screening the above loading and unloading positions, three trajectory datasets with relatively good quality are finally selected to test the algorithm. Figure 6 describes the data, including the visualization of the trajectory data, the travel time distribution, and the travel distance distribution.

To analyze the performance of the spatial anomaly detection of the algorithm, we manually mark all trajectories and then calculate the accuracy of the algorithm. The performance analysis of temporal anomaly detection uses the method of comparing the different window sizes.

Figures 7 and 8 show visualizations of spatiotemporal anomalies using a hot map. The hot map, which uses different colors to indicate statistically abnormal values, is generated by counting the number of anomalous grids in the window on a test trajectory. The red points are GPS points, which represent the travel route of the vehicle. Figure 8 displays only the nonabnormal routes.

4.2. Results of Spatial Anomaly Detection. In Figure 7, different spatial anomalies are highlighted. The normal trajectories are almost the shortest routes between the loading and unloading positions. An analysis of each picture is given below. In Figure 7(a), a route with more serious spatial anomalies chooses the East 4th Ring Expressway instead of the East 5th Ring Expressway. There are 106 trajectories in this dataset, 6 of which are on this route. Through map matching, it is found that these trajectories do not leave the East 4th Ring Expressway. Therefore, these abnormalities may be caused by the driver’s negligence, which leads to a failure to change lanes in time and a missed opportunity to enter the 5th Ring Expressway. In Figure 7(b), there are some locations with serious abnormalities and concentrated areas. This may be due to vehicles with temporary
Figure 5: Clustering results of the loading and unloading points (map comes from https://www.tianditu.gov.cn/).

Figure 6: Data description (map comes from https://www.tianditu.gov.cn/). (a) Visualization. (b) Travel time distribution. (c) Travel distance distribution.
transportation tasks. In Figure 7(c), the purpose of detouring vehicles may be to avoid large intersections. However, Figure 8(c) shows that there are not many temporal anomalies at the intersection. Instead, there is a temporal anomaly on the street near the destination. Therefore, it is reasonable that the vehicle is avoiding possible traffic jams on that street.

4.3. Results of Temporal Anomaly Detection. In Figure 8, red represents abnormally severe areas in terms of anomalies, and blue represents abnormally mild areas. From Figure 8, it can be concluded that most of the abnormal areas are close to the intersection, and the areas close to the starting and ending points are more likely to be abnormal.

In Figure 8(a), the temporal anomalies mainly occur at the intersection of Beijing Shenyang Road and Huosha Road and at the intersection of Beijing Shenyang Road and the airport expressway. A road with almost no temporal anomalies is the East 5th Ring Expressway. The temporal anomalies in Figure 8(b) mainly occur in the suburbs between Hebei Province and Beijing city and at the intersection of Guangming West Road and Xichang Road in Langfang city. In the suburbs between Hebei Province and Beijing, the roads pass through settlements, and temporal anomalies occur in dense residential areas. The intersection of Guangming West Road and Xichang Road in Langfang city has a large traffic volume and is prone to traffic congestion. Figure 8(c) shows the Shangqing Bridge in Beijing city. The figure shows that temporal anomalies mostly occur on the roads approaching adjacent residential areas and at intersections (such as the intersection with Shangqing Bridge), which is consistent with the actual situation.

The above analysis describes the reasons for spatio-temporal anomalies under different conditions. It is demonstrated that the anomaly detection algorithm proposed in this paper can address actual situations.

4.4. Results of Abnormal Trajectory Detection. Finally, a trajectory is used for qualitative analysis to present all the results of the algorithm. Figure 9 shows a trajectory with ID 27636. The spatial shape of this trajectory is shown in Figure 9(a). The blue points are normal, and the red points are abnormal.

Figures 9(b)–9(d) display the temporal anomalies, spatial anomalies, and risk value of the trajectory, respectively. Figure 9(d) expresses all the spatiotemporal anomaly information. A change in color indicates a change in the density of the temporal (Figure 9(b)) and spatial (Figure 9(c)) anomalies in $t_i$ of equation (1). From Figure 9(d), we can identify anomalous trajectory segments
and determine a level of risk to warn of the potential security risk. Figure 9(d) is the concept of risk value calculated according to equation (4). Users can define the risk level according to their own needs according to different value of risk. In equation (4), $t_1$ and $t_2$ can be adjusted according to the varied demand of different scenarios.

5. Analysis

This section comprehensively analyzes the impact of different parameters on accuracy and analyzes the differences between our proposed method and iBOAT and TRAOD algorithms. After that, the results of the three methods are explained on our data.

Environment: our algorithm is mainly implemented in Python. The machine used for the experiment has a 9th generation i7 CPU and 32 G memory. The operating system is Windows 10 x86_64, and the compiler is Spyder 4.15.

5.1. Spatial Anomaly Parameter Analysis. Spatial anomaly detection for vehicles focuses mainly on the impact of the grid size on accuracy and processing time. Figure 10 displays these results. In Figure 10, the blue line represents the impact of the grid size on the accuracy, and the red line shows the impact of the grid size on the processing time of the program. The x-axis is the grid size. The left y-axis is the number of normal trajectories. The right y-axis is the processing time. A representative dataset is selected considering the long transport distance of urban freight vehicles. There are 106 trajectories in this dataset, of which 99 are normal and 7 are abnormal. Figure 10 shows that when using a 250 m × 250 m grid, most normal trajectories were found, and the processing time was at a high level. Therefore, 250 m is selected for spatial anomaly detection.

5.2. Temporal Anomaly Parameter Analysis. For temporal anomaly detection, manual methods are not used, because it is difficult for manual methods to find a reasonable standard to judge when and where the GPS points are abnormal. Therefore, we observe the impact for other window sizes, with the result of window size = 2 as the benchmark, and we conduct a contrastive analysis to explain the feasibility and advantages of the algorithm proposed in this paper. Figure 11(a) shows the method, where the x-axis represents time and the y-axis represents space. Spaces 5 to 3 (the window is 2) are used as a benchmark to judge the algorithm’s performance under different parameters, e.g., spaces 5 to 1 (the window is 4). Next, we will analyze the performance and accuracy of WT-iBOAT.

Performance: Figure 11(b) shows the processing time of WT-iBOAT and iBOAT. The processing time of WT-iBOAT is approximately 2.5 times that of iBOAT. It does not change...
with the number of historical trajectories. This shows that the processing time is stable, and the processing time of each GPS point is approximately 0.006 seconds. Therefore, the algorithm can be used for real-time detection.

Accuracy: the travel time of a trajectory will fall into one of four cases: true positives (TP) are the number of anomalous travel times of trajectories that are classified as anomalous; false positives (FP) are the number of normal travel times of trajectories that are classified as normal; true negatives (TN) are the number of anomalous travel times of trajectories that are classified as normal; and false negatives (FN) are the number of normal travel times of trajectories that are classified as anomalous. We report these cases and calculate the common metrics of precision, true positive rate (TPR), defined as \( \text{TPR} = \frac{TP}{TP + FN} \), and true negative rate (TNR), defined as \( \text{TNR} = \frac{TN}{TN + FP} \). A perfect classifier will have TNR = 1 and TPR = 1.

In Figure 12(a), window size \( \leq 6 \), quantile = 75, TPR > 0.75; in Figure 12(b), quantile > 75, window size = 4, and TPR > 0.825. This reflects that as long as the parameter settings are reasonable, our algorithm can achieve relatively excellent results. To be more specific, when considering the effect of the window size on accuracy, the quantile is determined to be 75. This is because when the quantile is 75, there is a higher requirement for the travel time, which is convenient for comparing the results of the algorithms. The 75th quantile and the 2 windows are selected as the target values. Figure 12(a) plots the results with regard to different values of window size. TPR decreases rapidly with increasing window size, which indicates that the larger the window size is, the longer the travel time allowed by the window, so the proportion of positive sample detections decreases. Quantifying TPR, when the window sizes are 3, 4, 5, 6, 7, 8, 9, and 10, the TPR calculated by the algorithm is 87%, 82.5%, 78%,
77%, 75%, 74%, 72%, and 71% of the benchmark, respectively.

However, TNR hardly declines, indicating that the window has little impact on the judgment of negative samples, which is actually the role of the adjustment mechanism in the algorithm. The adjustment mechanism effectively reduces the impact of previous moments in the i-th window.

When studying the impact of the quantiles on accuracy, the window size is chosen as 4, because this is an obvious turning point in Figure 12(a). This shows that enough information is considered in 4 windows. Figure 12(b) shows that TPR and TNR increase with increasing quantiles. This means that the higher the tolerance is, the less temporal anomaly detection occurs. Specifically, the TPR of 4 windows and 70th quantile is 82% of that of 2 windows and 70th quantile; the TPR of 4 windows and 75th quantile is 82% of that of 2 windows and 75th quantile; the TPR of 4 windows and 80th quantile is 85% of that of 2 windows and 80th quantile; the TPR of 4 windows and 85th quantile is 87.5% of that of 2 windows and 85th quantile; the TPR of 4 windows and 90th quantile is 91.5% of that of 2 windows and 90th quantile; the TPR of 4 windows and 95th quantile is 95% of that of 2 windows and 95th quantile.

5.3. Abnormal Trajectory Detection and Analysis. In order to illustrate the algorithm more effectively, we compare our
algorithm with the TRAOD algorithm and iBOAT algorithm in the case study. TRAOD is an offline algorithm, which can get abnormal trajectories by clustering the segments of different trajectories [8]. The TRAOD algorithm has high efficiency on finding out the trajectories with more lane changes. The results are as follows: in 212 trajectories, our method found all 48 abnormal trajectories, while iBOAT algorithm only found 13 abnormal trajectories, and TRAOD algorithm found 39 abnormal trajectories. Among them, 11 of the 48 abnormal trajectories found by our method are the same as 11 of the 39 abnormal trajectories found by TRAOD, of which 4 belong to spatial anomaly and 7 belong to temporal anomaly. iBOAT and TRAOD have the same four spatial anomaly trajectories. All the 13 abnormal trajectories found by iBOAT algorithm are in 48 trajectories found by our method. The results are shown in Table 2.

The reason for getting such different results is that the principle of the TRAOD algorithm is different from that of our algorithm. TRAOD algorithm needs to cut the trajectory into segments and then classify the trajectory segments. Then, the TRAOD algorithm uses the weighted sum of vertical distance, parallel distance, and angular distance to calculate the distance between track segments. This makes the algorithm very sensitive to parameters, and the results of different parameters will be quite different.

The outliers obtained by our algorithm are mainly determined by the travel time of vehicles in the time window, while the outliers obtained by TRAOD algorithm are mainly determined by the distance between the trajectory segments. This leads our algorithm to pay more attention to the travel speed of vehicles, and the TRAOD algorithm to pay more attention to the anomaly caused by lane change.

At the same time, a serious disadvantage of the TRAOD algorithm is that it takes a long time to adjust parameters and needs different parameters for different data. In our three datasets, we use three groups of different parameters to judge the trajectory anomaly of the TRAOD algorithm. This may highlight the advantages of our method with respect to TRAOD algorithm in practical use.

### 6. Discussion

In this section, we discuss the role of the adjustment mechanism in the algorithm. Then, an example is used to explain how equation (1) improves the results of spatial anomaly detection.

### 6.1. Overall Discussion of the Adjustment Mechanism

For the overall discussion of the adjustment mechanism, the speed ratio can explain the information of the window well. If speed ratio $> 0$, the travel time is abnormal. If speed ratio $< 0$, the travel time is normal. The larger the speed ratio is, the more serious the temporal anomaly, and the smaller the speed ratio is, the milder the temporal anomaly. The following case needs to be explained: when the test trajectory immediately turns a spatial anomaly into a temporal anomaly, the travel time of the historical trajectory is zero, and the speed ratio is infinite. To detect this, when calculating the speed ratio, greater sensitivity is required. Therefore, the historical speed is taken as 50 m/s, so the first speed ratio is equal to $(50-250)/50 = 0.5$, and the subsequent speed ratio will also be greater than 0.5.

Figure 13 is the kernel density diagram of the abnormal speed ratio of all datasets. The green, yellow, black, and blue areas are the kernel density diagrams of the 75th quantile with a window of 4, the 90th quantile with a window of 4, the 75th quantile with a window of 8, and the 90th quantile with a window of 8, respectively. The first row is the kernel density diagram without the adjustment mechanism, and the second row is the kernel density diagram with the adjustment mechanism.

Note that the different color areas in the first row are more dissimilar than their counterparts in the second row. This shows that the adjustment mechanism makes the abnormal speed ratio more reasonable in different situations. The adjusted results tend to be more concentrated in areas with smaller speed ratios, and the larger areas are reduced accordingly, which is a manifestation of the rationality of the adjustment.

### 6.2. Improved Analysis of Spatial Anomaly

The algorithm judges whether the test trajectory is a spatial anomaly by calculating the support values. However, simply using the support values causes errors due to the grid. Figure 14 shows the results of the problem and improvement. The different colors in the left figure represent different support values. Red represents larger support values, and blue represents smaller support values. In the green box, when the grid size is $250 \times 250$ meters, a normal spatial point may be judged as an abnormal point. Section 4 equation (1) can be used to solve this problem. The right figure shows the result of using this formula when $t_2$ is 10 minutes and $t_1$ is 3 minutes. The right figure reflects the correction of the calculation results very well. Blue indicates a low abnormality, and red indicates a high abnormality.
7. Conclusion

Taking trajectory security as the research background, this paper provides an effective real-time risk monitoring scheme for detecting the anomalous trajectory of logistics vehicles. The main idea is to combine the spatial and temporal anomaly detection by using the time probability distribution of historical trajectory and taking the fixed position window as the detection fragment.

The results show that the algorithm can achieve higher accuracy, and the risk detection reaches 82.5% of the benchmark on our dataset, outperforming the traditional iBOAT algorithm (60%). At the same time, the false positive rate (FPR) of iBOAT is 0.031, and the FPR of our method is 0.044. Comparing the FPR values of the two methods, the higher FPR value of our method is caused by the supplemented temporal anomaly detection. However, we successfully found 22.5% more anomalies than iBOAT algorithm by using adjustment mechanism in temporal anomaly, so that the distribution of temporal anomaly

results calculated by our method is more practicable as explained in Section 6.1. It is fairly acceptable that the FPR is increased by only 0.013 with 22.5% more anomalies correctly detected. Therefore, the result proves that our proposed algorithm can effectively detect the abnormal trajectories of urban freight vehicles in real time with not only higher accuracy, but also lower false alarm rate. This can provide more efficient, faster, and safer logistics express services for addressing unexpected trajectory anomaly events.

There are still some deficiencies in the current research. For example, when the historical trajectory data set is large, the real-time query of historical data will cost a lot of calculation, and the probability model will help solve this problem. Further research can establish probability model or neural network for different data to improve the speed of discovery from different start and end points.

In general, (1) the framework we use can be reasonably and widely applied to security service scenarios for urban freight vehicles. (2) For temporal anomalies, we propose a simple, fast, and efficient real-time temporal anomaly
discovery method. (3) The proposed method provides idea for further research.

Data Availability
The trajectory data used to support the findings of this study may be released upon application to the Beijing Jiaotong University and the cooperative logistics company, who can be contacted at zhlwei@bjtu.edu.cn.

Conflicts of Interest
The authors declare that they have no conflicts of interest.

Acknowledgments
The research reported in this paper was supported by grants from National Natural Science Foundation of China (Grant no. 91746201) and National Key R&D Program of China (2018YFB1601600).

References
[1] B. Jean-Pierre, "Profile of the perpetrator of the Nice terror attack that took place on 14th July 2016: a terrorist whose modus operandi may have been imitated in other European attacks, annales médico-psychologiques," Revue Psychiatrique, vol. 176, no. 6, pp. 607–612, 2018.
[2] M.-R. Estela and H. Assumpção, "How safety affects destination image projected through online travel reviews," Journal of Destination Marketing and Management, vol. 18, Article ID 100469, 2020.
[3] D. F. Xie, T. L. Zhu, and Q. Li, "Capturing driving behavior heterogeneity based on trajectory data," International Journal of Modeling, Simulation, and Scientific Computing, vol. 11, no. 3, pp. 199–329, 2020.
[4] Z. He, L. Zheng, P. Chen, and W. Guan, "Mapping to cells: a simple method to extract traffic dynamics from probe vehicle data," Computer-Aided Civil and Infrastructure Engineering, vol. 32, no. 3, pp. 252–267, 2017.
[5] J. Chen, K. Qi, and S. Zhu, "Traffic travel pattern recognition based on sparse global positioning system trajectory data," International Journal of Distributed Sensor Networks, vol. 16, 2020.
[6] Y. Li, Q. X. Huang, M. Kerber, L. Zhang, and L. Guibas, "Large-scale joint map matching of GPS traces," in Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 214–223, ACM, Orlando, FL, USA, 2013.
[7] A. Bera, S. Kim, and D. Manocha, "Realtime anomaly detection using trajectory-level crowd behavior learning," in Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1289–1296, Las Vegas, NV, USA, July 2016.
[8] J. G. Lee, J. Han, and X. Li, "Trajectory outlier detection: a partition-and-detect framework," in Proceedings of the IEEE International Conference on Data Engineering, April 2008.
[9] X. Li, J. Han, S. Kim, and H. Gonzalez., "ROAM: rule- and motif-based anomaly detection in massive moving object data sets," in Proceedings of the 2007 SIAM International Conference on Data Mining, Minneapolis, MN, USA, April 2007.
[10] X. Zixuan and W. U. Wei, "Anomaly detection by exploiting the tracking trajectory in surveillance videos," Science China Information Sciences, vol. 63, no. 5, pp. 197–199, 2020.
[11] Z. Daqing, L. Nan, Z. Zhi-Hua, C. Chao, S. Lin, and L. Shijian, "IBAT: detecting anomalous taxi trajectories from GPS traces," in UbiComp, Proceedings of the 13th international conference on Ubiquitous computing (UbiComp’11), Association for Computing Machinery, pp. 99–108, New York, NY, USA, 2011.
[12] Z. Jie, J. Wei, L. An, L. Guanfeng, and Z. Lei, "Effective and efficient trajectory outlier detection based on time-dependent popular route," World Wide Web, vol. 20, no. 1, pp. 111–134, 2017.
[13] C. Chao, Z. Daqing, C. Samuel, L. Nan, S. Lin, and L. Shijian, "Real-time detection of anomalous taxi trajectories from GPS traces," in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 63–74, Springer, Berlin, Germany, 2011.
[14] Z. Fu, W. Hu, and T. Tan, "Similarity based vehicle trajectory clustering and anomaly detection," IEEE International Conference on Image Processing, vol. 2, pp. 2–602, 2005.
[15] L. Liping, C. Hao, and X. Shengwu, "Distributed anomaly detection algorithm for spatio-temporal trajectories of vehicles," in Proceedings of the 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/ICC), pp. 590–598, Guangzhou, China, December 2017.
[16] L. Song, R. Wang, D. Xiao, X. Han, and C. Shi, "Anomalous trajectory detection using recurrent neural network," in Advanced Data Mining and Applications, Springer, Berlin, Germany, 2018.
[17] H. Wu, W. Sun, and B. Zheng, "A fast trajectory outlier detection approach via driving behavior modeling," in Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 837–846, New York, NY, USA, November 2017.
[18] C. ZaiBen, S. HengTao, and Z. Xiaofang, "Discovering popular routes from trajectories," in Proceedings of the 2011 IEEE 27th International Conference on Data Engineering, pp. 900–911, IEEE Computer Society, Hannover, Germany, April 2011.
[19] S. Jani, W. Jan, and O. Juha, "Generating heat maps of popular routes online from massive mobile sports tracking application data in milliseconds while respecting privacy," ISPRS International Journal of Geo-Information, vol. 11, pp. 1813–1826, 2015.
[20] W. Luo, H. Tan, L. Chen, and M. Lionel, "Finding time period-based most frequent path in big trajectory data," in Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, pp. 713–724, Association for Computing Machinery, New York, NY, USA, June 2013.
[21] S. Treerapot, S. Agachai, and H. W. Ho, "Statistical estimation of freight activity analytics from global positioning system data of trucks," Transportation Research Part E: Logistics and Transportation Review, vol. 140, Article ID 101986, 2020.
[22] L.-Y. Wei, Yu Zheng, and W.-C. Peng, "Constructing popular routes from uncertain trajectories," in Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 195–203, Association for Computing Machinery, New York, NY, USA, August 2012.
[23] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: algorithms and applications," The VLDB Journal, vol. 8, no. 3, pp. 237–253, 2000.
[24] J. Lee and N.-W. Cho, "Fast outlier detection using a grid-based algorithm," *PLoS One*, vol. 11, no. 11, Article ID e0165972, 2016.

[25] X. Wang, Y. Zhang, H. Liu, Y. Wang, L. Wang, and B. Yin, "An improved robust principal component analysis model for anomalies detection of subway passenger flow," *Journal of Advanced Transportation*, vol. 2018, Article ID 7191549, 12 pages, 2018.

[26] N. Suzuki, K. Hirasawa, K. Tanaka, Y. Kobayashi, Y. Sato, and Y. Fujino, "Learning motion patterns and anomaly detection by human trajectory analysis," in *Proceedings of the 2007 IEEE International Conference on Systems, Man and Cybernetics*, pp. 498–503, Montreal, Canada, October 2007.

[27] H. Yin, Z. Hu, and X. Zhou, "Discovering interpretable geosocial communities for user behavior prediction," in *Proceedings of the 2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pp. 942–953, Helsinki, Finland, May 2016.

[28] M. A. M. Ali and M. Ismail, "Abnormal driving detection using real time global positioning system data," in *Proceedings of the 2011 IEEE International Conference on Space Science and Communication (IconSpace)*, pp. 1–6, Johor Bahru, Malaysia, 2011.

[29] L. Sun, D. Zhang, C. Chen, P. S. Castro, S. Li, and Z. Wang, "Real time anomalous trajectory detection and analysis," *Mobile Networks and Applications*, vol. 18, no. 3, pp. 341–356, 2013.

[30] 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*, pp. 422–431, New York, NY, USA, August 2014.

[31] B. Yingyi, C. Lei, W.-C. Ada, and L. Dawei, "Efficient anomaly monitoring over moving object trajectory streams," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 159–168, Association for Computing Machinery, New York, NY, USA, 2009.

[32] L. Yang, M.-P. Kwan, X. Pan, B. Wan, and S. Zhou, "Scalable space-time trajectory cube for path-finding: a study using big taxi trajectory data," *Transportation Research Part B: Methodological*, vol. 101, no. jul, pp. 1–27, 2017.

[33] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: why and how you should (still) use DBSCAN," *ACM Transactions on Database Systems*, vol. 42, no. 3, pp. 1–21, 2017.