IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 13, 2020, accepted February 1, 2020, date of publication February 7, 2020, date of current version February 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2972299

Abnormal Trajectory Detection Based on a Sparse Subgraph

XUJUN ZHAO, YUANQI RAO, JIANGHUI CAI, AND WENQIANG MA
School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China

Corresponding author: Jianghui Cai (jianghui@tyust.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572343, Grant U1731126, and Grant U1931209, in part by the Natural Science Foundation of Shanxi Province of China under Grant 201901D111257, and in part by the Taiyuan University of Science and Technology Scientific Research Initial Funding under Grant 20192013.

ABSTRACT Traditional abnormal trajectory detection algorithms mainly involve the measurement of a single feature; however, the influence of other features on abnormal trajectory is ignored, resulting in the inability to fully discover the abnormal trajectory in the trajectory database. To overcome this limitation, we propose an abnormal trajectory detection method – called *TADSS* – to find the hidden abnormal trajectory by using a comprehensive measurement. Firstly, we employ three kernel functions to measure the time, velocity and position feature values of trajectory data, where the kernel functions extract semantic feature of the position, time feature of trajectory, and velocity feature of object motion from each trajectory data. Secondly, we propose a feature fusion strategy to measure the similarity of trajectory data, where we assign weights to the above kernel functions and then use a linear combination approach to fuse the weighted kernel functions. Thirdly, we build a trajectory feature graph by using the above fused kernel functions, and then divide the trajectory feature graph into a plurality of subgraphs by using a conventional graph clustering technique. Last, we propose a sparse subgraph method to detect abnormal trajectory, where a novel weight coefficient concept is used to distinguish sparse subgraph. Experimental results driven by both the vehicle trajectory data of Shanghai city and the Atlantic hurricane data demonstrate the performance of our *TADSS*.

INDEX TERMS Trajectory data, abnormal detection, kernel function, sparse subgraph.

I. INTRODUCTION

With the development of positioning devices, wireless network, video monitoring and infinite sensors, large amounts of trajectory data are collected from a variety of devices. One of the most common types of trajectories is generated by *GPS* equipped vehicles. Other types of trajectories probably can be generated by smart phones, online checkin data, geotagged messages or media in social networks, RFID readers, and other such applications. The moving objects can be human beings, animals, vehicles, and even natural phenomena (e.g. hurricanes). Furthermore, these trajectory data contain a large amount of valuable information. Currently, in many fields, such as those of unmanned driving systems, navigation systems, and intelligent traffic system, research on trajectory data receiving increasing attention.

Abnormal data represent objects that are extremely different or inconsistent with the other data objects in a data set [1]. An anomalous trajectory is a trajectory that has a large local or global difference with most of the other trajectories, in terms of similar metrics [2]. In the hurricane detection by a meteorological station, the abnormal changes in the hurricane trajectory can be captured, and the casualties and property losses can be reduced by providing people with an early warning; When using the application of an online car-hailing service, the passengers can receive a warning from the application if the driver takes a detour or deviates from the normal driving route. In this manner, the passengers can avoid paying more than required and avoid incidents [3], [4]. Therefore, it is of considerable practical significance to search for an abnormal trajectory in a large trajectory data set. Most of the existing methods measure the similarity in terms of the location property of the trajectories [5], [6]. However, in real trajectory data sets, the trajectory data involve many different features. The attributes of the trajectory data are effectively mined by trajectory clustering algorithms, for example, cluster analysis of the trajectory point density around a place to mine the attributes of the place [7],

The associate editor coordinating the review of this manuscript and approving it for publication was Xin Luo.

and the selection of the cluster center effectively improves the quality of the cluster [8]. To realize a more comprehensive mining of the abnormal trajectory data, we propose a trajectory similarity measure model that include semantic feature of the position, time feature of trajectory, and velocity feature of object motion. In addition, we also design an abnormal trajectory detection scheme by searching the sparse subgraph.

A. MOTIVATIONS

The abnormal trajectory detection in this study is motivated by the following three observations:

- Trajectory similarity measurement is an important step in abnormal trajectory detection, where traditional methods only measure the similarity from a single dimension. There is no doubt that these measurement methods lose some important information, which makes the detection of abnormal trajectory inaccurate, or even unable to detect the real abnormal trajectory effectively.
- The abnormal trajectory detection is an expensive operation, because traversal of data points on different trajectories is time consuming. The high overhead of the traversal operation becomes more pronounced when it comes to large datasets. Each trajectory dataset contains many trajectory objects, each of which contains a large number of data points. When we directly compare the data points pertaining to different trajectory data, the corresponding tasks and running time will increase significantly.
- The existing abnormal trajectory detection methods require the setting of a large number of parameters, which leads to a higher influence of human factors in the experimental conclusions.

Motivation 1: The trajectory data contain the position, speed, and time attributes. In the measurement of the similarity of trajectories, if the similarity originates from a single information measurement, the detection results tend to ignore a large amount of the hidden trajectory information. For example, the similarity of the trajectory in terms of the position attributes cannot indicate whether the data are abnormal in terms of speed attributes. Similarly, we can detect whether a vehicle is overspeeding by using the velocity attribute of the trajectory data; however, we cannot detect the abnormal behavior in time attribute in this manner. Hence, in the traditional abnormal trajectory detection methods, as the metrics correspond to a single feature, the abnormal behavior the other attributes cannot be accurately detected.

Motivation 2: A trajectory data set contains many trajectories, and each trajectory contains dozens to thousands of data points. In the detection of abnormal trajectories, a large amount of time is required to traverse data points on different trajectories. The traditional detection algorithms may cause the problem of insufficient memory due to the direct operation of the data points. These algorithms need to be recalculated when the parameters are modified, which requires a large amount of time. At the same time, some data points are repeatedly calculated for each test, which causes a lot of time consumption. Therefore, the traditional abnormal trajectory algorithm cannot be used in massive data sets.

Motivation 3: To set the parameters, the traditional methods detect the abnormal trajectories by dividing the trajectory data into subsegments and calculating the proportion and density of the abnormal trajectory subsegments [9]. For instance, Lee JG proposed the *TRAOD* algorithm framework [10], in which many parameters are required to be set. These artificial parameters may affect the results of the experiment. A larger number of parameters correspond to a larger interference of the human factors. Therefore, the use of fewer parameters can make the results more objective.

B. OUR APPROACH AND CONTRIBUTIONS

In this paper, we propose a method, sparse-subgraph-based abnormal trajectory detection using multiple information sources (i.e., *TADSS*). *TADSS* consist of three distinct phases:

- Establishment of trajectory similarity measure model: the important features of the trajectory are reflected in many aspects, such as semantic feature of the position, time feature of trajectory, and velocity feature of object motion, and so on. In first phase, the feature information of each trajectory is extracted from position, velocity and time by using multiple kernel functions.
- Calculation and fusion of the weighted similarity measures: we assign weights to each feature vector and use a linear combining approach for feature fusion of multiple vectors. The fused feature values are used to measure the similarity of two trajectories.
- Anomalous trajectory detection: the fused feature values are mapped to a feature graph, and then the graph is divided into a plurality of subgraphs by using a conventional graph clustering technique. The anomalous trajectory are obtained by searching the sparse subgraphs.

Our key contributions are summarized as follows.

1) Novel trajectory similarity measure: we employ three kernel functions to measure the time, velocity and position feature values of trajectory data. First, we propose a model for extracting the semantic features from the trajectories. In the model, a semantic kernel is applied to solve the problem of data misalignment when the coordinates of the entire trajectory are measured. Second, we design a novel speed-based similarity metric that effectively accommodates the velocity properties of trajectory data with features of position and time labels. This approach allows us to identify the same vector data at different locations.
2) Multi-feature fusion mechanism for trajectory similarity detection: We propose an evaluation mechanism for abnormal trajectory using multiple feature vectors. First, we assign different weights to each kernel function, in which the selection of weights can be based on domain knowledge or mining tasks. Second, we use a linear combination approach to fuse weighted kernel functions. In our method, the behavior of different feature

vectors can be represented approximately by a fusion vector, in which different feature vectors are jointly learned to maximize the consistency of all the diffusion processes. The fusion mechanism can be applied to different application scenarios only by adjusting the corresponding weight combination.

3) Abnormal trajectory detection: We propose a sparse-subgraph-based abnormal trajectory detection algorithm *TADSS*. First,we build a trajectory feature graph by using the above fused kernel functions, in which a trajectory is treated as a node in the graph. Second, we divide the trajectory feature graph into a plurality of subgraphs by using a conventional graph clustering technique. Third, we propose a sparse subgraph method to detect abnormal trajectory, where a novel weight coefficient concept is used to distinguish sparse subgraph (see Definition 6).

4) Experiments on multiple real datasets: We use the vehicle trajectory data of Shanghai city to test the performance of our algorithms *TADSS*, and apply the Atlantic hurricane data to evaluate the impacts of parameters on the experimental results. Our experimental results show that *TADSS* accurately detect real abnormal trajectories, and outperform the existing algorithm in terms of efficiency.

C. ROADMAP

The rest of the paper is structured as follows: Section II introduces the related work on abnormal trajectory research. Section III proposes a method of the feature fusion measurement by using the kernel functions and introduces abnormal trajectory detection process in sparse subgraph. In section IV, *TADSS* algorithm is introduced and analyzed. Section V describes experimental settings and offers result analysis. Finally, we conclude our study in section VI.

II. RELATED WORK

There exist many research directions for the trajectory data. Currently, researchers are focusing on optimizing effective trajectory indexing structures [11], and developing methods for trajectory frequent pattern based on grid sequence [12]–[14], trajectory outlier detection based on trajectory information entropy distribution [15], abnormal trajectory detection for intelligent transport system [16], trajectory uncertainty management [17], [18], and mining knowledge from trajectory data [19], [20], etc. Among these domains, the study of abnormal trajectories is an important research direction. The main methods pertaining to abnormal trajectory detection mainly include classification based methods, cluster based methods, density based methods, and statistics based methods. In this section, we also introduce the existing abnormal trajectory detection algorithm.

A. DISTANCE BASED TRAJECTORY DETECTION METHOD

The algorithms based on distance measurement are a type of classical detection method. These algorithms mainly measure the similarity of the trajectories by calculating the distance between two data points, such as the Euclidean distance.

The concept of an abnormal trajectory was originally proposed by Knorr *et al.* [21] These researchers proposed a method for the abnormal trajectory detection based on the distance, and indicated that the trajectory data mainly have four features: position information (longitude and latitude), magnitude of the velocity, direction of the velocity and timestamp. In addition, these researchers studied the position properties of the trajectories and detected the abnormal trajectories by using traditional positional measures. When considering a complete trajectory, the disadvantage of this method is that the abnormal fragments can not be detected because the errors are evenly distributed to each object.

To overcome this problem, Lee *et al.* [10] proposed a segmentation detection framework algorithm named *TRAOD*. This algorithm involves two steps: First, the algorithm conducts a coarse grained partitioning for the trajectory data, and later, coarse grained detection is performed. Second, the tracjectory data are divided into fine grains to detect an abnormal trajectory. In this algorithm, the similarity is determined by considering the Hausdorff distance [27] in pattern recognition.

The abovementioned abnormal trajectory algorithm based on distance measurement considers only the position feature of the trajectory data, and other features of the trajectory data are ignored. To overcome this limitation, Yuan *et al.* [22], [28] proposed a trajectory outlier detection algorithm based on similar structures, which considered other features of the trajectory data; this method made the results more objective and more similar to a real situation.

B. DENSITY BASED TRAJECTORY DETECTION METHOD

In the distance based abnormal trajectory algorithms, the user must specify the global distance threshold which is difficult to select. The abnormal trajectories can be regarded as global outliers detected by the global distance threshold. However, some data sets exhibit features related to the density of the data objects, which can be regarded as local abnormal trajectories.

To overcome the shortcomings of the distance based methods, Liu *et al.* [23] proposed a density-based trajectory outlier detection algorithm (DBTOD). The DBTOD employ considers the concept of the trajectory density of the neighborhood object distribution. This algorithm considers two aspects: the distance between the segments, and the number of segments within a given range. Compared with the distance based algorithm, this method overcomes the problem of the parameter sensitivity of the distance based methods. To improve the efficiency of the algorithm, Liu *et al.* [24] used an R-tree to store the trajectory data to accelerate the algorithm.

C. CLASSIFICATION BASED TRAJECTORY DETECTION METHOD

Li *et al.* [25] proposed an anomaly trajectory detection algorithm, named Roam, based on the classification. This method

TABLE 1. Comparisons of trajectory detection algorithm.

Solutions	Measurement method	Multi-feature	Data reduction	Number of artificial parameter	Abnormal detection
TADSS (This study)	Kernel-based	Yes	Yes	2	Yes
DB-outlier [21]	Distance-based	No	No	2	Yes
TRACLUS [9]	Distance-based	No	No	2	No
TRAOD [10]	Distance-based	No	Yes	6	Yes
TOD-SS [22]	Distance-based	No	No	4	Yes
DBTOD [23]	Density-based	No	No	6	Yes
RTOD [24]	Density-based	No	Yes	6	Yes
Roam [25]	Classification-based	No	No	2	Yes
SKSS [26]	Kernel-based	No	No	2	No

mainly consists of three parts: First, we divide the trajectory into a unit and construct a feature space with the relevant attributes. Second, we determine the hierarchy of features through the automatic extraction. Third, we build a classifier to detect the effective region in the feature space.

Roam is a supervised abnormal trajectory detection algorithm, and the trajectory data are difficult to find in the tagged data. As a result, it is not possible to find the results of interest to the observer for many common datasets. Furthermore, it is difficult to manage the local abnormal data processing of the trajectories.In contrast to the traditional trajectory measurement methods mentioned above, Ramirez-Padron R *et al*. [26] introduced similarity kernels for the nearest neighbor based on the outlier detection and introduced a method for the classification of similar kernel functions. Hamid *et al*. proposed an elite ensemble framework to manage the two control parameters of the proposed algorithm [29]. Fernando *et al*. [30] proposed a novel model based on deep learning to predict the future motion of a pedestrian given a short history of their, and their neighbours, past behaviour.

Table 1 summarizes the other methods associated with the *TADSS*: (1) The *TADSS* uses the kernel function dimensionality reduction trajectory data and a measure of the feature fusion to make the results more relevant to the observers; (2) the DBOutlier uses the distance to measure the similar trajectories and subsequently finds the anomalous trajectory data; (3) the TRACLUS involves a clustering algorithm framework; (4) the *TRAOD* uses the Hausdorff distance to measure the similarity of the trajectories and involves a detection framework for the abnormal trajectories; (5) the TODSS uses a similar structure to detect the abnormal trajectories and considers other feature of the trajectories; (6) the DBTOD is a density based trajectory anomaly detection algorithm, which enables the observer to better set the threshold; (7) the RTOD uses the R tree to store the data to make the algorithm more efficient; (8) Roam detects the anomalous trajectories by classifying the tagged data objects; (9) the SKSS introduces a method to detect the abnormal data by using kernel functions.

In summary, the traditional outlier trajectory detection algorithm directly processes the data position information and adopts the metric of a single feature of the trajectory data, ignoring the other feature information inclued in the trajectory information. However, the trajectory data contain other features such as the velocity information, and different observers are interested in different trajectory data characteristics. For example, the vehicle trajectory position information is the same on a road segment. However, if the data exhibit considerably higher values than those of other vehicle trajectories, vehicle overspeeding or emergency events (involving fire vehicles, ambulance vehicles, etc.)may be occurring. Such information cannot be detected if only one feature is measured from the location information. Abnormal trajectory algorithms using the entire trajectory data as the basic unit, have not been extensively relatively studied. Based on the graph based clustering method, the kernel function can be used to map the trajectory data to the high dimensional space to ensure that the data can be linearly divided; subsequently, the trajectory data are constructed as an undirected weighted graph. The kernel function is used to measure the similarity between the nodes in the graph, and the subgraph obtained by the graph clustering is processed through spectral clustering. Finally, the anomalous trajectory is searched in the subgraph. The trajectory data have a more uncertain distribution, and thus, we use the spectral clustering to obtain better clustering results.

In this paper, the trajectory data are mapped into a graph model, and the trajectory anomaly detection process is illustrated in Fig.1. The main process of the trajectory anomaly detection process is divided into three parts: In the first part, we use the corresponding kernel function method to measure the similarity of the trajectory data in terms of three features; in the second part, we assign different weights to the metrics of the corresponding features and add them to obtain the similarity measure of the feature fusion. In the third part, we divide the graph model into subgraphs through spectral clustering and search the subgraphs for abnormal data.

Table 2 lists the major notations used throughout this paper.

III. MULTI FEATURE FUSION TRAJECTORY DATA METRIC AND ANOMALY DETECTION

In this section, we map the trajectory data into an undirected graph model. Each trajectory is treated as a vertex in the graph model, which means that we consider a trajectory data as an object, and the similarity measure of different trajectories is

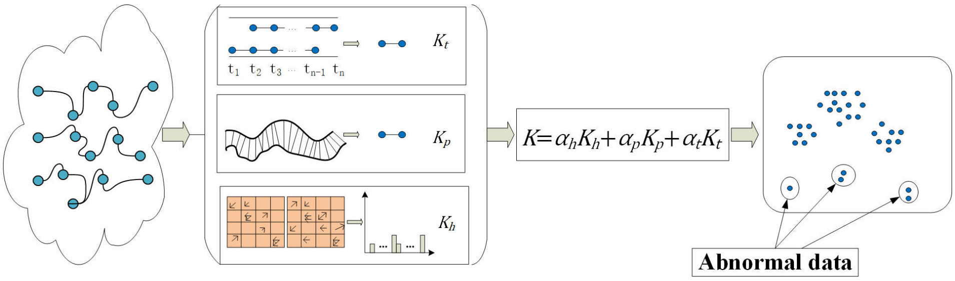


FIGURE 1. The description of the trajectory anomaly detection process.

TABLE 2. Symbol and notation.

Symbol	Notation	Symbol	Notation
X	trajectory data set	x_k	the object in X
n_k	the element of x_k	N	object number of X
G	graph model of X	G'	subgraph of G
V	vertices set of G	V'	vertices set of G'
E	edge set between V	E'	edge set between V'
ω	weight of edge E	ω'	weight of edge E'
$K(T_i, T_j)$	measurement of feature fusion	$c_n(\pi_m(i))$	coordinate of n-th trajectory data x_n
$d_{(i,j)}$	Euclidean distance of $c_n(\pi_m(i))$	K_p	measurement under position feature
$\mathcal{L}_l$	SPM kernel in l level	K_h	measurement under velocity feature
K_t	measurement under time feature	$\|V\|$	number object of V
$M(G')$	weight coefficient	τ	threshold of $M(G')$

represented by the weight value between the vertices. Since each trajectory is composed of many data points, we need to reduce the dimensionality of the data to ensure that we can view a trajectory as an object. In this work, we reduce the dimensional trajectory data by using kernel functions, which map the eigenspace of the samples to an eigenspace with higher dimensions [31]. Consequently, the results show many new features that could not be observed before, because the data are mapped through the kernel functions.

The trajectory data often contain different characteristic attributes, and the measurement methods of these different attributes cannot be unified. For example, for the location characteristic attribute, because the number of points contained in different trajectory data is different, we need to stretch the data to align the data. For the velocity characteristic attribute, we need to consider the velocity information of the trajectory with the position attribute, and we need to compare the velocity magnitudes and the directions at different positions. Ahmad Nazari1 *et al*. proposed a weighting framework to weight up the better clusters (clusters with higher qualities) [32], and Frouzan Rashidi *et al*. proposed a weighted metric in the graph model [33]. In this manner, the different features can be measured using a weighted fusion metric.

Since the trajectory data are mapped to a high dimensional space through the kernel function, it is difficult to distinguish the distribution of the mapped data. The traditional clustering algorithms, such as the k-means algorithm, find it difficult to adapt to this phenomenon. Thus, in this work, we adopt the spectral clustering method. Compared with the traditional comparison algorithm, it is better to distinguish, extract and amplify the useful features through the combination of the kernel function mapping and spectral clustering. This approach can obtain the correct clustering results even if the traditional clustering algorithm cannot obtain satisfactory results.

A. DESCRIPTION OF THE MULTI FEATURE FUSION TRAJECTORY AND FEATURE INFORMATION MODEL

In this paper, the trajectory data are mapped into vertices in the graph, and the trajectory data are formalized as follows:

Definition 1: We define a trajectory data set: $X = \{x_1, \ldots, x_k, \ldots, x_n\}^T$. *The trajectory data element* x_k *is an element composed of* n_k *consecutive points,* $n_k = \{s_k, v_k, e_k\}$. s_k *represents the position feature information,* v_k *represents the speed feature information, and* e_k *represents the time feature information.*

Definition 2: Given N Trajectories, the trajectory data are mapped into the graph G with N vertices, where G is denoted as $G = (V, E, \omega)$. $V = \{v_1, v_2, \ldots, v_n\}$ *represents the set of vertices for the trajectories;* $E = \{e_1, e_2, \ldots, e_n\}$ *represent the edges for the vertices;* $\omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$ *represents the weight that corresponds to the similarities between two trajectories.*

For the graph model for the trajectory data, the vertex similarity is measured by the multiple information feature fusion technique. In this paper, we use a similarity matrix $K(T_i, T_j)$ to represent the similarity of the vertices in the graph model. We construct a similarity matrix K, where $\omega = K$, which means that each element of K represents the similarity ω of the vertices in the graph [34]:

$$K(T_i, T_j) = \sum_{1}^{m} \alpha_m K_m(T_i, T_j) \tag{1}$$

$K(T_i, T_j) = 1$, where $T_i = 1, 2, \ldots, N$ and $T_j = 1, 2, \ldots, N$, which means that a trajectory is most similar to itself. α_m represents preassigned weights, which are used to reflect the users' interest in the m dimensional features.

When considering the similarity of different trajectory location features, we obtain different trajectory data containing different numbers of points due to the different object movements and data collectors, etc. The measurement of the similarity of different trajectories inevitably leads to the misalignment of data and unmatched trajectory data with a different number of points. To solve this phenomenon of data misalignment, we use the global alignment kernel (GAK) [35] to measure the similarity between the different trajectory data.

Definition 3: We denote the coordinate set of the $n-th$ *trajectory as* $c_n(\pi_m(i))$*;* $\pi(i)$ *represents the* $i-th$ *point of the trajectory.* $c_1(\pi_m(i) = \{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i)\}$ *represents the coordinate set of the* $1-th$ *trajectory, and* $\pi_1(x)$ *represents* x_1*, which is the x coordinate of the* $1-th$ *coordinate.*

We compare with two time series, $(c_1(\pi(i)), c_2(\pi(j)))$, which represent the two dimensional coordinates of the two different trajectories. Because the number of points from the two sequences is not identical, we adopt the idea of the DTW: First, we obtain the coordinate points from the two trajectories and calculate the Euclidean distance between the two coordinate points by using formula (2). We store the distance into the matrix d.

$$\begin{cases} d_{(m1,m2)} = {d_x}^2 + {d_y}^2. \\ d_x = (c_1(\pi_{m1}(x)) - c_2(\pi_{m2}(x))), \\ d_y = (c_1(\pi_{m1}(y)) - c_2(\pi_{m2}(y))). \end{cases} \tag{2}$$

$$d_{(i,j)} = e_{ij} + min(d_{(i-1,j)}, d_{(i,j-1)}, d_{(i-1,j-1)}) \tag{3}$$

Second, we use formula (3) to calculate the distance of the dynamic time warping (DTW) between the two sequences. Let us assume that $\varphi(x, y)$ is equal to $d_{(x,y)}$, which is the DTW distance between the two trajectories of $d_{(x,y)}$. Finally, we build the kernel function using the global alignment kernel [36]:

$$K_p(T_i, T_j) = e^{-\varphi(x,y)} \tag{4}$$

By calculating the position information of the trajectory data using formula (4), we can effectively solve the problem of misalignment between the two position information sequences of the trajectory.

When we deal with the real trajectory data, the velocity of the trajectory at different positions have different meanings. For example, the instantaneous velocity of a certain region is concentrated in a relatively smaller interval, which may be an intersection or a school road segment. If the direction is concentrated in a small range of angles, then this section may prohibit the turnaround, among other phenomena. The traditional abnormal trajectory algorithms cannot identify an abnormal instantaneous velocity or direction existing in the data set because they measure the location feature of the trajectory data. In addition, the velocity information packet of the trajectory data contains the magnitude and direction of the velocity, and the velocity position at this time should be considered. The traditional methods of measuring the vector data cannot the process the vector velocity and position information at the same time. As a result, the spatial pyramid kernel function method used in image processing is adopted in this paper to measure the similarity of the velocity features of the trajectories.

The speed information is gradually divided into different orders of magnitude. First, the velocity information is divided into 2^l level region blocks by the position coordinates, $l = 0, 1, \ldots, n$. The number of direction and speeds values is counted separately in each area. The speed magnitude and direction are divided into different regions, and the number of directional angles and velocity magnitudes having the same region are counted separately into different histograms h_k. Next, the size of the histogram h_k in the same region is considered to perform similarity matching. Comparing the two histogram sizes of the two trajectories in this region, the minimum of the two values is taken as the matching similarity measure of the portion. Finally, the histograms in the different regions for the same magnitude are subjected to an accumulation operation to obtain the similarity measure under the magnitude.

$$\mathcal{L}_l(h_k(T_i), h_k(T_j)) = min(h_k(T_i), h_k(T_j)) \tag{5}$$

Formula (5) is used for the calculation. The area block similarity of the 2^{l+1} level is computed by repeating the above operation until the area is divided into the expected levels.

$$K_h(T_i, T_j) = \sum_{i=0}^{2} \frac{1}{2^i}(\mathcal{L}_{l+1} - \mathcal{L}_l) \tag{6}$$

Finally, the spatial pyramid kernel is calculated using formula (6), and we regard the calculated results as the similarity in the velocity features.

Next, we introduce the method to measure the trajectory data under the time features. $t(0)$ and $t(1)$ represent the beginning and end times of the trajectory, respectively. After determining the difference between these two times, the difference in the time measurement is compared using formula (7) to determine $t(T_i, T_j)$. Finally, the radial basis (RBF) kernel function is used to calculate the measure similarity through formula (8) to determine $K_h(T_i, T_j)$. Considering the N trajectory, the parameter β is obtained as $\frac{1}{\sum_0^N \sum_0^N t(T_i,T_j)}$.

$$t(T_i, T_j) = \sum_{i=0}^{1} (t_k(i) - t_k(j))^2 \tag{7}$$

$$K_t(T_i, T_j) = e^{-\beta t(T_i,T_j)} \tag{8}$$

Different observers may be interested in different aspects of the same trajectory data set. For example, when a passenger takes a taxi, he/she is concerned about whether the driver takes a detour. When private car owners are traveling, they are concerned about whether the vehicle is overspeeding. In the current time frame, it is important for emergency vehicles to avoid congested roads. Therefore, we designed a weighted fusion measure to accommodate the different requirements of the different observers.

$$K(T_i,T_j)=\alpha_p K_p(T_i,T_j)+\alpha_h K_h(T_i,T_j)+\alpha_t K_t(T_i,T_j) \quad (9)$$

We initialize the values of α_p, α_h and α_t as $\alpha_p = \frac{K_p}{\sigma(K)}, \alpha_h = \frac{K_h}{\sigma(K)}, \alpha_t = \frac{K_t}{\sigma(K)}$, and these values satisfy the conditions of formula (10).

$$\begin{cases} \alpha_p+\alpha_h+\alpha_t=1, & 0\le\alpha_p,\alpha_h,\alpha_t\le 1. \\ \sigma(K)=\sigma(K_p)+\sigma(K_h)+\sigma(K_t) & \end{cases} \quad (10)$$

$\alpha_p, \alpha_h, \alpha_t$ respectively represent the weights of $K_p(T_i, T_j)$, $K_h(T_i, T_j)$, $K_t(T_i, T_j)$.

For some same trajectory data, different experts pay close attention to different features. For example, some experts are only interested in semantic feature of the position, however, other experts are interested in time feature of trajectory. In order to suit the demands of various experts, we may modify the parameter values in formula (9) to highlight the importance of some trajectory features. If experts pay attention to a single trajectory feature, we set the weight of this feature to 1 and the weight of other features to 0. If experts pay attention to multiple features, we assign an average weight to these features where the sum of the weights is 1. For example, if an observer pays attention to the route of the taxi, then his/her attention demands for the position feature. We consider the position measurement weight value as 1, and the measurement weight values of speed and time as 0 (i.e., $\alpha_p = 1$, $\alpha_h = 0$, and $\alpha_t = 0$). If an observer pays attention to the abnormality of driving process, he/she should be concerned about the abnormal trajectories under the position feature (detour) and speed feature (overspeeding). we consider the measurement weight values of position and speed as 1, and the time measurement weight value is 0 (i.e., $\alpha_p = 0.5$, $\alpha_h = 0.5$, and $\alpha_t = 0$).

We make $w_{(T_i,T_j)} = K(T_i, T_j)$, $w_{(T_i,T_j)} \in W$. The measurement values of the various trajectory similarities are calculated and stored in the upper triangular matrix. According to the fact that the similarity matrix is a symmetry matrix, we obtain the similarity matrix W as follow:

$$W = \begin{bmatrix} \omega_{11} & \cdots & \omega_{1n} \\ \cdots & \cdots & \cdots \\ \omega_{n1} & \cdots & \omega_{nn} \end{bmatrix} \quad (11)$$

B. ABNORMAL DETECTION BASED ON SPECTRAL CLUSTERING

In this paper, we use spectral clustering method to divide the node objects in the graph model. The result of the clustering is that the objects within the same group are similar, and the objects are not similar if they come from different groups. In the graph model, the objects in the graph are divided into several subgraphs through spectral clustering, which makes the objects remarkably similar if they come from the same subgraph, whereas the objects in different subgraphs have an incredibly low similarity.

Definition 4: Given graphs $G =< V, E, \omega >$ and $G' =< V', E', \omega' >$, if $V' \subseteq V$, $E' \subseteq E$, $\omega' \subseteq \omega$, we consider that graph G' is a subgraph of G.

Definition 5: Given a subgraph G', two different trajectories T_i, T_j, and the weight $\omega_{(T_i,T_j)}$ between T_i and T_j, the weight coefficient is denoted as $M(G')$, which is a ratio between the sum of all trajectories weight $\sum_{h=1}^{|E|} \omega_{(T_i,T_j)}$ and the number of $\mid E \mid$. Formally, we have

$$M(G') = \frac{\sum_{h=1}^{|E|} \omega_{(T_i,T_j)}}{\mid V \mid} \quad (12)$$

In particular, when a subgraph contains a single vertex, we consider the weight coefficient of the subgraph to be 0.

Definition 6 (Sparse Subgraph): Given a weight coefficient threshold τ, the subgraph G' and its weight coefficient $M(G')$, If $M(G') \le \tau$, then G' is a sparse subgraph. Please note that the value of τ is much smaller than the average of the weight coefficients of all subgraphs.

We also describe the computational process of the data processing, which consists of five steps:

Step 1: Each trajectory is mapped into a node in the graph by considering definition 1. We calculate the similarity measure between the different trajectory vertices by using formula (9) $K(T_i, T_j) = \alpha_p K_p(T_i, T_j) + \alpha_h K_h(T_i, T_j) + \alpha_t K_t(T_i, T_j)$.

Step 2: We construct the similarity matrix W, and insert the similarity measurements of different trajectories into this matrix; $w_{(T_i,T_j)} = K(T_i, T_j)$, $w_{(T_i,T_j)} \in W$.

Step 3: We build a degree matrix $D_{(i,j)} = \sum_1^n w_{(T_i,T_j)}$. That is, $D_{(i,j)}$ is expressed as the sum of every row in the W matrix, and $L = D - W$ is constructed.

Step 4: We solve for the eigenvalue k in the matrix L. In this paper, the difference between the eigenvalues $K = k_{(i+1)} - k_i$ is calculated. If the values of $K_{(i+1)}/K_i$ and $K_i/K_{(i-1)}$ change significantly, the number of clusters is selected as K in this paper.

Step 5: The k-means algorithm is used for the clustering of the feature vectors, and the trajectory data nodes are divided into each subgraph G'. Finally, we calculate the weight coefficient for each subgraph G' by $M(G') = \frac{\sum_{h=1}^{|E|} \omega_{(T_i,T_j)}}{|V|}$, and output the weight coefficient $M(G') \le \tau$ graph of the trajectory data.

IV. ALGORITHM IMPLEMENTATION

In this section, we describe the implementation of the algorithm. For the abnormal trajectory detection algorithm, we mainly use a measurement method of the feature fusion and algorithm for the abnormal trajectory detection of the sparse subgraph. The method of feature fusion measurement mainly consists of three kernel functions: DTW_Kernel, Spatial_Pyramid_Kernel and RBF_ Kernel. The detection algorithm mainly divides the subgraph through spectral clustering and later detects the abnormal trajectories in the subgraph by judging that the weight coefficient is less than the threshold τ.

Algorithm 1 *TADSS*:Abnormal Trajectory Detection of Sparse Subgraph

```
Input: Trajectory data;
Output: abnormal trajectory;
 1: n ← input the number of trajectories in the trajectory set
 2: for (i = 0; i < n; i++) do
 3:     $K_p \leftarrow$ DTW_Kernel;
 4:     for ($l$ = 0; l < 3; $l$ + +) do
 5:         $\mathcal{L} \leftarrow$ Spatial_Pyramid_Kernel($l$);
 6:     end for
 7:     $K_h \leftarrow \sum_{i=0}^{2} \frac{1}{2^i}(\mathcal{L}_{l+1}(h_k(i), h_k(j)) - \mathcal{L}_l(h_k(i), h_k(i));$
 8:     $K_t \leftarrow RBF\_Kernel;$
 9:     $K(T_i, T_j) = \alpha_p K_p(T_i, T_j) + \alpha_h K_h(T_i, T_j) + \alpha_t K_t(T_i, T_j);$
10: end for
11: Spectral_Clustering();
12: if $\frac{\sum_{h=1}^{|E|} \omega(T_i, T_j)}{|V|} \leqslant \tau$ then
13:     $Outlier \leftarrow$ V;
14: end if
15: return $Outlier$
```

Algorithm 2 *DTW_Kernel*: Kernel Function DTW

```
Input: position information (longitude and latitude coordinate position of the trajectory);
Output: location similarity measure $K_p$;
 1: $n_1, n_2 \leftarrow$ Number of position information points in trajectory 1 and trajectory 2
 2: for (i = 0; i< $n_1$; i++) do
 3:     for (j = 0; j< $n_2$; j++) do
 4:         $Dis = (x_1 - y_1)^2 + (x_2 - y_2)^2;$
 5:         $D_{i,j} \leftarrow Dis;$
 6:     end for
 7: end for
 8: for (i = 0; i< $n_1$; i++) do
 9:     for (j = 0; j< $n_2$; j++) do
10:         $d_{(i,j)} = e_{ij} + min(d_{(i-1,j)}, d_{(i,j-1)}, d_{(i-1,j-1)});$
11:         $\varphi_{(n_1,n_2)} \leftarrow Dis$
12:     end for
13: end for
14: $DTW\_Kernel \leftarrow e^{-\varphi_{(n_1,n_2)}};$
15: return $DTW\_Kernel$
```

The TADSS consists of the following five steps:

Step 1: We measure the similarity of the trajectory data in terms of the position attribute. The location similarity measure is calculated by calling the DTW kernel function (see line 3 in algorithm 1) by using algorithm 2. First, we calculate the distance between two points on two different trajectories using the Euclidean distance (see line 4 in algorithm 2) and store the distance between these two points in the distance matrix (see line 5 in algorithm 2). Second, we search for the shortest distance between the two trajectories through the idea of the dynamic time warping (see line 10 in algorithm 2). Finally, the kernel function is used to calculate the similarity of the two trajectories (see line 13 in algorithm 2).

Step 2: We calculate the size of the histogram bin of each trajectory and calculate the similarity measure of the velocity (see lines 4-7 in algorithm 1). Since the velocity characteristic is a vector data, we use the spatial pyramid kernel function to perform the measurement in algorithm 3. First, we store the velocity information into a dictionary, where the key value corresponds to the coordinates of the trajectory, and the value is the magnitude and direction of the trajectory velocity (see line 2 in algorithm 3). Second, we divide the coordinate information of the trajectories into small grids (see lines 3 and 4 in algorithm 3) and count the number of velocities and directions in each grid; this information is stored in histogram bin (see lines 5–12 in algorithm 3). Third, we compare the velocities by using the spatial pyramid kernel function and introduce the specific calculation formula (6) (see line 7 in algorithm 1) in the detection algorithm.

Step 3: For the time information, we calculate the time similarity measure of the trajectory by calling the radial basis kernel function (see line 8 in algorithm 1). We may focus on the trajectory for a given period of time. We calculate the distribution of the trajectory interval by using algorithm 4. The difference between the beginning and end times of the trajectory is calculated(see line 3 in algorithm 4), and the similarity of the time data is later calculated by using the radial basis kernel function (see line 4 in algorithm 4).

Step 4: Previously, we introduced the three kernel functions in the measurement. Next, we introduce the abnormal trajectory detection algorithm. According to definition 1, we treat the trajectory data set as a graph model. Next, we set the weights by observing the degree of interest of the observer and obtain a measure of the feature fusion (see line 9 in algorithm 1). Finally, we divide the subgraph through spectral clustering and calculate the weight coefficient of the subgraph. Next, we output the abnormal trajectories by judging if the weight coefficient is less than the threshold τ (see lines 11–14 in algorithm 1).

Through the above description of the algorithm, it can be noted that the proposed algorithm shortens the detection time by using the kernel function method. In addition, if the new data keep increasing, we only need to add the calculation for the new data on the basis of the original stored similarity matrix. Compared with the traditional algorithm, we do not need to repeat the calculation for the previous data.

V. EXPERIMENTAL RESULTS AND ANALYSIS

This section describes the experimental evaluation of the *TADSS* and its comparison with the *iBOAT*, the *TPRO* and the *TRAOD*; details regarding the *iBOAT*, the *TPRO* and the *TRAOD* can be found in [10], [13], [14]. For all the reported results, the experimental environment was as follows: Intel(R) Core (TM) i5-6640HQ CPU, 8GB. Windows 10. The *TADSS* algorithm was coded in Python (python 3.6).

Algorithm 3 *SPM_Kernel*:Spatial Pyramid Kernel

```
Input: speed information of trajectory;
Output: speed similarity measure $K_h$;
$n_1, n_2 \leftarrow$ speed information of trajectory 1 and trajectory 2;
Convert the trajectory data to $dic_1$ and $dic_2$ (with coordinate information as key value, speed size and direction as value);
gird()$\leftarrow$ Generate a matrix containing the positional information of the two trajectories on the map coordinates;
The gird matrix is divided into several matrix regions with a size of $2^l$ and the size of the matrix after segmentation is $len * wid$;
for (i = 0; i< $n_1$; i++) do
    for (j = 0; j< $n_2$; j++) do
        $Bin_1$ $\leftarrow$Count the magnitude and direction of the velocity in area 1;
        $Bin_2$ $\leftarrow$Count the magnitude and direction of the velocity in area 2;
        Sim = min($Bin_1$, $Bin_2$);
        Sum+ =Sim;
    end for
end for
return Sum
```

Algorithm 4 *RBF_Kernel*: Radial Basis Kernel Function

```
Input: time information of trajectory;
Output: time similarity measure $K_t$;
$t_1(start), t_1(end) \leftarrow$ Start time and end time of trajectory 1
$t_2(start), t_2(end) \leftarrow$ Start time and end time of trajectory 2
sim = $((t_1(start) - t_2(start))^2 + (t_1(end) - t_2(end))^2)^{\frac{1}{2}}$;
RBF_Kernel = $e^{-sim}$;
return Sim
```

We test the abnormal detection on a taxi trajectory data set and hurricane data set.

In this paper, to verify the feasibility of the algorithm, the data set we used corresponded to February 20, 2007; the Shanghai taxi trajectory data in the data set contain more than four thousand taxi vehicle trajectories within 24 h and the movement trajectory with a vehicle sampling interval of 1 min. The trajectory data attributes included the ID number of the vehicle, vehicle longitude dimension information, timestamp information, and magnitude of the instantaneous velocity. The vehicle angle is a radian of the vehicle direction and the North direction. The trajectory data sets have different numbers of trajectories, and each trajectory has approximately about 1700 to 7000 points. We organized the trajectory data set in terms of the location of the properties of the latitude (longitude), speed features (longitude latitude information, magnitude of the instantaneous velocity, vehicle orientation), and time (time stamp). Because of the considerable amount of information, we considered a point every 20 min. Two experiments for the data sets were performed in this paper, and each data set contained 82 different paths. The hurricane data of the year 2017–2018 were used for testing and included 28,798 data points with 1131 trajectories. We tested the influence of the algorithm parameters and compared the experiment results with those of the *TRAOD*.

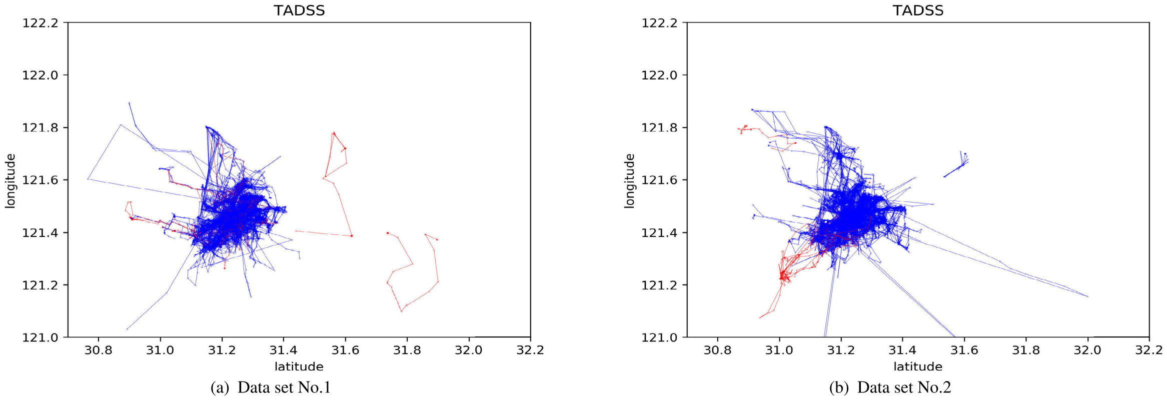
Throughout our extensive experiments, the parameters are set as follows. In the accuracy evaluation experiment, the weight coefficient threshold is set to 0(i.e., $\tau = 0$,), and the number of cluster is set to 20(i.e., $k = 20$). In the efficiency evaluation experiment, we evaluate the impacts of parameter k(i.e., the number of cluster) on the mining efficiency of *TADSS*, so the number of cluster is varied from 20 to 200. Specifically, the various k are 20, 30, 40, 50, 60, 80, 90, 100, 105, 200, respectively.

A. AUTHENTICITY OF THE EXPERIMENTAL RESULTS

To verify the authenticity of the algorithm, we divided the data set into two sets. We tested the rental vehicle considering different parameters in the experiment. We set the number of cluster as 20 and 35 in the first and second data sets, namely, data sets No.1 and No.2, respectively. Next, we performed different measurements to note the differences in the results.

Fig.2 shows the abnormal trajectory detection under the velocity attribute, where $\alpha_h = 1$. We examine five abnormal trajectories in the No.1 data set and twelve abnormal trajectories in the No.2 data set. The experimental results are shown in 2(a) and 2(b), respectively. Compared with the abnormal trajectories detected under the location attribute, it is difficult to intuitively see that some trajectories in Fig.2 are abnormal trajectories, which may be caused by the overspeeding. It is difficult for us to intuitively visualize the anomalies of these data under the position features of the trajectory. These anomalies are often overlooked when using the other trajectory detection algorithms.

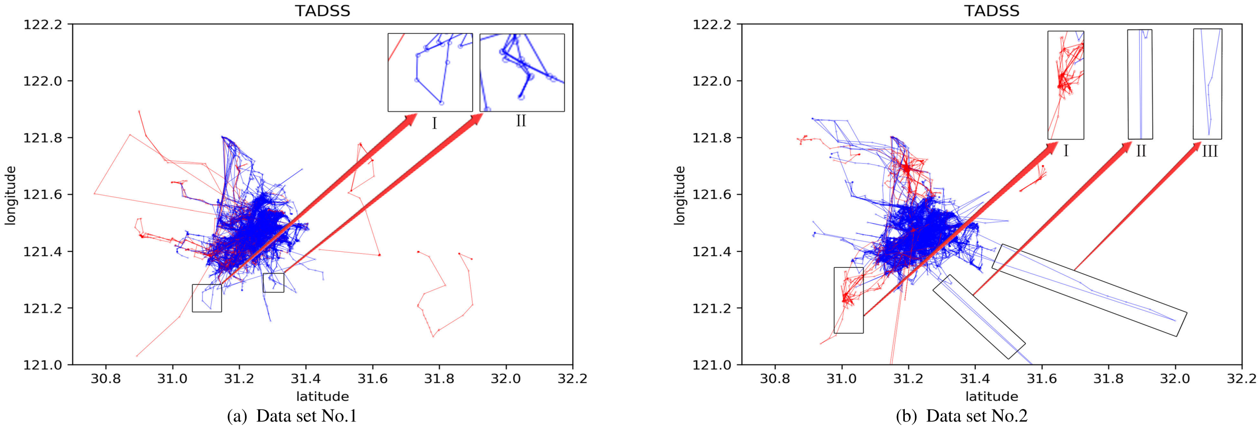
In Fig.3, $\alpha_p = 1$. This aspect means that we are testing for the abnormal trajectories under the location property of the data set. The location features of the abnormal trajectories are shown in Fig.3(a), which indicates that there exist eight abnormal trajectories in the No.1 data set; Fig.3(b) shows that eleven abnormal trajectories exist in the No.2 data set. Intuitively, we can see from Fig.3 that the trajectory marked in red is an obvious abnormal trajectory, which deviates from most of the trajectories in terms of the position features. However, in the second data set, there exist two distinct trajectories that have not been detected. The reason for this phenomenon may be that the DTW kernel function detects only the global data, which may ignore some local abnormal trajectories. When the location feature is adopted to measure the trajectory data, although we can detect the majority of the abnormal trajectories, the local abnormal information is often ignored.



(a) Data set No.1

(b) Data set No.2

FIGURE 2. Features of the velocity. (a) corresponds to dataset No.1 and (b) corresponds to dataset No.2. The abnormal trajectories are represented by red lines and the normal trajectories are represented by blue lines.



(a) Data set No.1

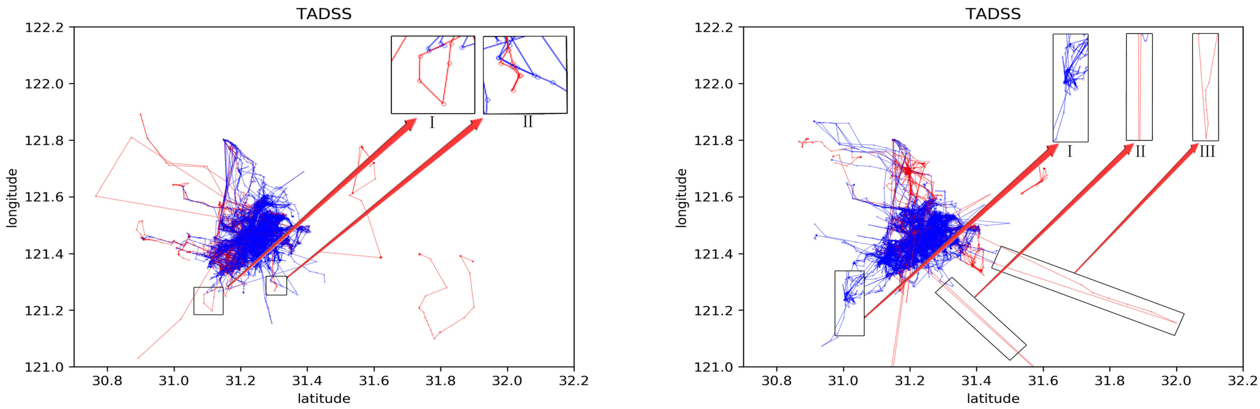
(b) Data set No.2

FIGURE 3. Features of the position. (a) corresponds to dataset No.1 and (b) corresponds to dataset No.2. The abnormal trajectories are represented by red lines and the normal trajectories are represented by blue lines.

From these experiments, it is easy to see that different measurements lead to different results. Different observers have different interests; therefore, we need a measure that can be adjusted. Consequently, our approach is an efficient algorithm for different observers. However, for the location properties of the trajectories, our method involves a global comparison. Next, we use a measure of the feature fusion. In Fig.4, we use a measure involving the fusion of the location and velocity features; $\alpha_p = 0.5$, $\alpha_h = 0.5$. Fig.4(a) shows that eleven abnormal trajectories exist in the No.1 data set; Fig.4(b) shows that fourteen abnormal trajectories exist in the No.2 data set. Comparing the results with previous experiments, it can be noted that more significant abnormal data is observed after the feature fusion.

We extract the local information, as shown in Figs. 3 and 4, which is reflected in several areas in the upper right of the figures. We mark these areas with Roman numerals such as I, II, and III. By observing areas I and II in Figs.3(a) and 4(a), we can observe that the metric through feature fusion can help detect more anomalies at the edges. At areas II and III in Fig.4(b), we can see two notable anomalous trajectories (i.e., the red lines), which are normal trajectories (i.e., the blue lines) in Fig.3(b). In contrast, some trajectories (i.e., the red lines) plotted in area I in Fig.3(b) are misjudged as anomalous trajectories, which can be detected correctly, as shown in area I in Fig.4(b). The phenomenon observed in Figs.3(b) and 4(b) can be attributed to the metrics after feature fusion being more accurate than the metrics before the fusion. From the above description, we can see that the degree of abnormality of areas II and III is significantly greater than that of area I, which indicates that the fused anomaly is more meaningful. These abnormal trajectories may be caused by the wrong location detected using the GPS, the driver taking a detour or the vehicle overspeeding.

In Fig.4(a), we found not only abnormal trajectories under the location characteristics but also several new abnormal phenomena at the edge of the data set. In Fig.4(b), we can clearly see the two abnormal trajectories, which were not detected when using the location feature. We find that the global kernel function may involve losses in the local abnormal trajectory, but this aspect is compensated by detecting the abnormal velocity. It can be seen that the velocity measurement can also compensate for the deficiency of the DTW kernel function in the local detection. In addition, several



(a) Data set No.1

(b) Data set No.2

FIGURE 4. Fusion features of position and velocity. (a) corresponds to dataset No.1 and (b) corresponds to dataset No.2. The abnormal trajectories are represented by red lines and the normal trajectories are represented by blue lines.

new abnormal trajectories at the edge of data set are found. Therefore, the measurement after the feature fusion makes the experimental results more valuable and proves that our method is meaningful.

B. INFLUENCE OF THE PARAMETERS

No vector data for the velocity are available in the hurricane data; therefore, we performed experiments only on the feature of the position. We detected 1,131 trajectories of the hurricane. After a considerable amount of implementation, we found that when we set the threshold value to as 0, we could obtain some abnormal trajectory data, and these results revealed some interesting phenomena. When we set different parameters, we obtained different abnormal trajectories, and the results are presented in the Table 3.

TABLE 3. Number of abnormal trajectories when using different parameters.

No.	Number of clusters	Number of abnormal trajectories
1	20	21
2	30	42
3	40	44
4	50	40
5	60	82
6	80	84
7	90	90
8	100	105
9	105	144
10	200	208

The experimental results are not difficult to interpret: When we set the threshold value as 0, the number of abnormal trajectories increases with the number of clusters.The possible reasons for this phenomenon are as follows: First, for large trajectory data sets, because there are many subgraphs with only one vertex, we regard the weight coefficients of these subgraphs as 0. Second, an increase in the number of clusters leads to an increase in the number of subgraphs, and the trajectory data are thus allocated to more subgraphs. Due to the abovementioned reasons, more sparse subgraphs are generated because the distribution of the trajectory data is more dispersed.

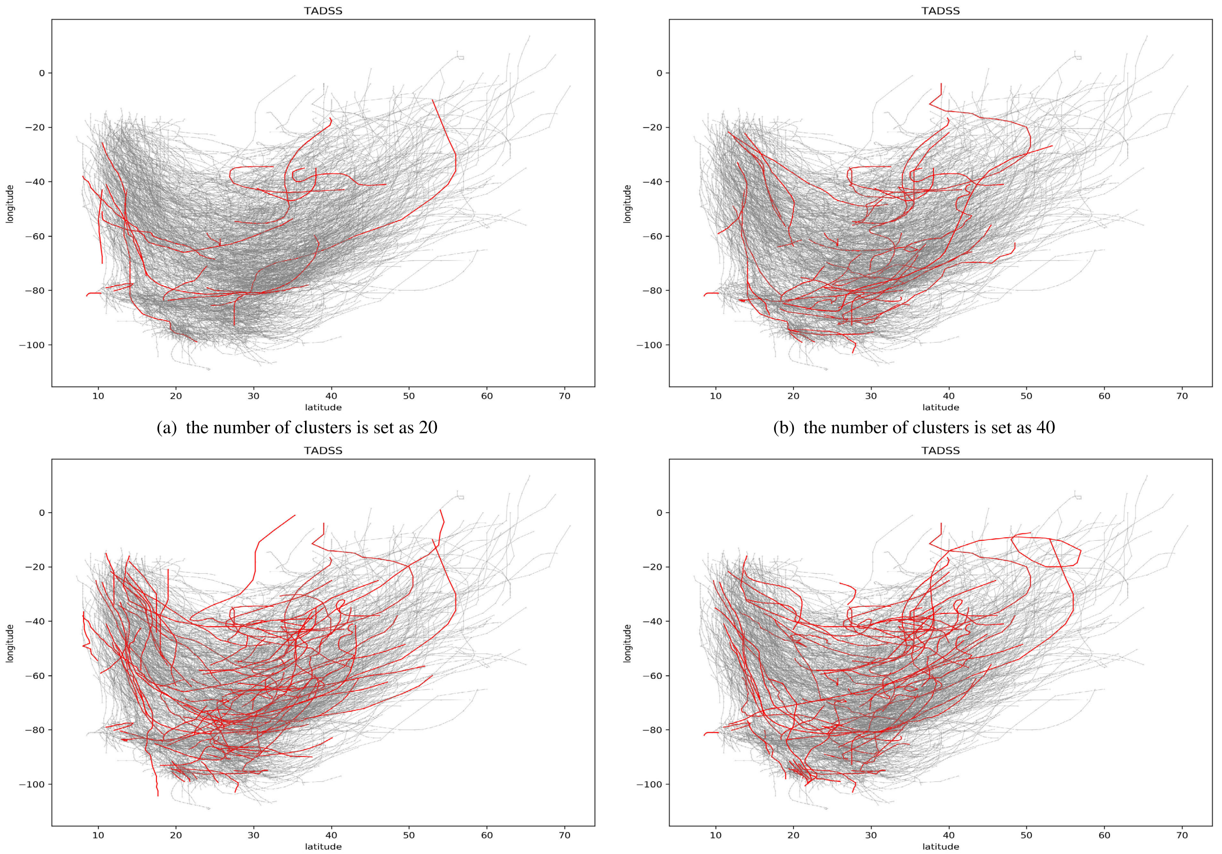
Next, we show a part of the results to determine the change trend of these results. We select the detection results for when the number of clusters is set as 20, 40, 60 and 75. The results of these tests are shown in Fig. 5.

Intuitively, as shown in Fig.5, we found that with the increase in the number of clusters, more abnormal trajectories could be found at the edge of the hurricane data. In Figs.5(a) and 5(b), these abnormal trajectories appear in the middle. As we continue to increase the number of classes in the cluster, the increase in the number of classes leads to the increase in the number of sparse subgraphs, because the trajectory data are allocated to more subgraphs. As a result, as the number of exceptions increases, more abnormal trajectories are present at the edges in Figs.5(c) and 5(d).

C. ALGORITHM COMPARISON

During the experiment, when the data correspond to 1131 trajectories, the *TRAOD* algorithm cannot be implemented due to memory reasons, and the *TADSS* algorithm can detect the abnormal trajectories. We performed the testing considering 404 trajectories. The *TRAOD* algorithm was tested with the following parameters settings: $D = 15.0$, $P = 1.0$, $F = 0.1$, WeightPar $= 5$, WeightAngle $= 5$, WeightPer $= 5$. The parameter settings for the *TADSS* algorithm are as follows: $k = 35$, $\tau = 0$. In the parameter setting of the algorithm, the *TADSS* algorithm sets two parameters, and the *TRAOD* algorithm sets five parameters. The advantages of the *TADSS* algorithm are thus notable.

The results of comparison of the two algorithms are shown in Fig.6. *TRAOD* found 30 anomalous trajectories and *TADSS* found 41 anomalous trajectories. It can be seen from the comparison that the anomaly trajectories found by the two algorithms have many distinct parts. It can be seen from Fig.6(b) that the *TADSS* algorithm finds the abnormal data that are out of position at marks 1 and 2; the most common



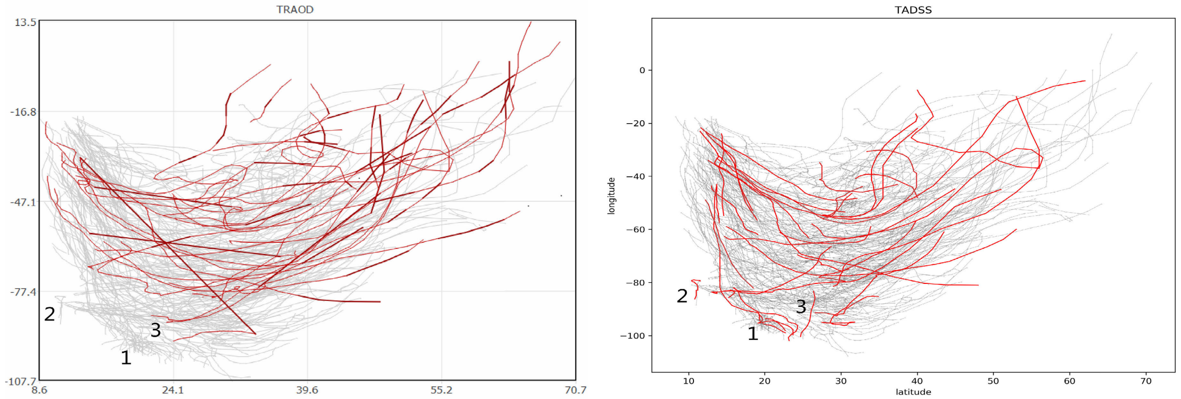
(a) the number of clusters is set as 20

(b) the number of clusters is set as 40

(c) the number of clusters is set as 60

(d) the number of clusters is set as 75

FIGURE 5. **Parameter influence experiment. (a) the number of clusters is set as 20; (b) the number of clusters is set as 40; (c) the number of clusters is set as 60; (d) the number of clusters is set as 75.**



(a) TRAOD

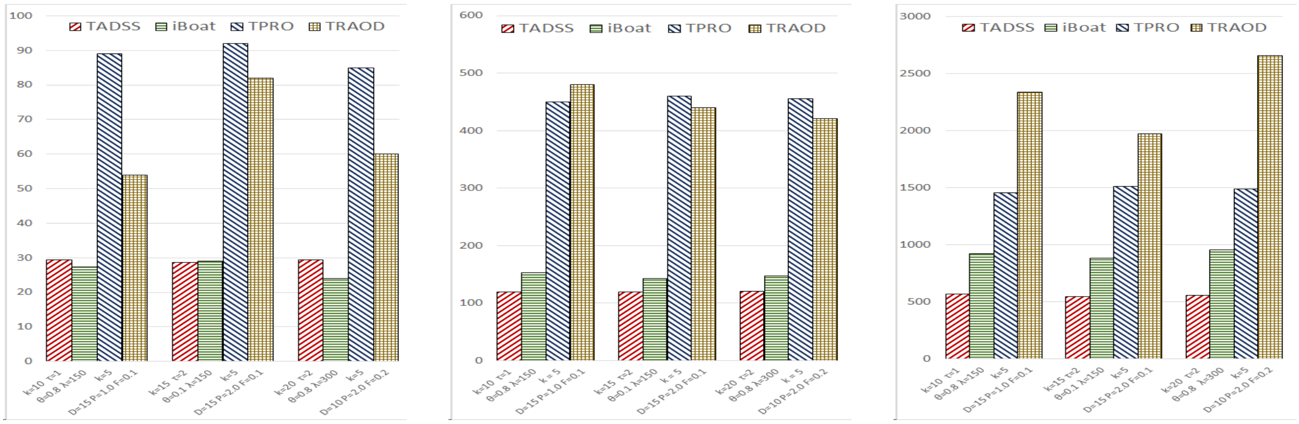
(b) TADSS

FIGURE 6. **Comparison experiment between TRAOD algorithm and TADSS algorithm.**

trajectory direction is the East West trend at mark 3, whereas the trend is a North South trend, which is an obvious abnormal data point. In Fig.6(a), *TRAOD* did not find any anomalous data at the bottom left of the figure, indicating that the *TADSS* algorithm is valid.

We set different parameters in order to test the effect of the parameters on the efficiency of the algorithm, which are listed in Table 4, and we also evaluate the efficiency of the four algorithms in the various trajectory data sets, where the trajectory numbers are 100, 200 and 403, respectively.

Fig.7. shows that *TADSS*'s efficiency behaves better than *iBOAT*, *TPRO* and *TRAOD* with the same number of the trajectory. In algorithm *TRAOD*, we need set two parameters for detecting abnormal trajectory. The first parameter is used to



(a) The number of 100

(b) The number of 200

(c) The number of 400

FIGURE 7. Comparison experiment efficiency between TRAOD algorithm and TADSS algorithm.

TABLE 4. Parameters setting of TADSS, iBOAT, TRPO and TRAOD.

$TADSS$	$iBOAT$	$TPRO$	$TRAOD$
k=10 τ=1	θ=0.8 λ=150	k=5 Δt=400	D=15 P=1.0 F=0.1
k=15 τ=2	θ=0.1 λ=150	k=5 Δt=200	D=15 P=2.0 F=0.1
k=20 τ=2	θ=0.8 λ=300	k=3 Δt=400	D=10 P=2.0 F=0.2

filter the anomalous coarse values; and the second parameter is utilized for obtaining the anomalous trajectories. In the coarse-grained detection stage, if a small amount of data is filtered by the *TRAOD*, then a large amount of data needs to be processed in the second stage. Hence, *TRAOD*'s mining efficiency measured in terms of running time is significantly reduced. In algorithms *iBOAT* and *TPRO*, an important parameter(i.e., grid size) needs to be adjusted many times, which will consume a lot of times during the process of grid establishment. The size of the grid affect the efficiency of the algorithm. Here, we set the grid size to the number of trajectory. In other words, one grid contains one point. On the contrary, *TADSS* algorithm can directly measure the global trajectory data to reduce the number of comparisons and the running time. The second observation is that the impact of the different parameters on *TADSS*'s running time is very weak when we test the same trajectory data set. Unfortunately, the impact of various parameters on *TRAOD* algorithm is significant. For example, in Fig.7(a), the running times of *TADSS* are 29.4s, 28.7s, and 29.4s, respectively, the running times of *iBOAT* are 27.3s, 29s, and 24s, respectively; the running times of *TPRO* are 89s, 92s, and 85s, respectively; and the running times of the *TRAOD* are 54s, 82s, 64s, respectively. Similarly, we test the running times of two algorithms in the number of 200 and 403 datasets, and the experimental results plotted in Fig.7(b) and 7(c) are consistent with those in Fig.7(a). The experimental results are shown as follows: In a small-scale data set, the running time of *TADSS* algorithm is close to *iBOAT*, and less than *TPRO* and *TRAOD* algorithms. As the number of trajectories is a large value, *TADSS* has a higher efficiency than *iBOAT*, *TPRO* and *TRAOD* algorithms.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose an abnormal trajectory data detection algorithm based on a sparse subgraph. The algorithm measures trajectory data from the multiple feature, which effectively solves the problem of single point measurement in traditional algorithm. Mining data from multiple features makes it easier to discover information hidden in large amounts of trajectory data. Moreover, the *TADSS* algorithm can effectively reduce the problem of multi parameter setting in traditional methods. The observer assigns the different weights to the different trajectory features according to their own interests, which makes it easier for the observer to find the abnormal trajectory under their interested features. Finally, we test with the real data. The next work is to parallelize *TADSS*, which will greatly improve the efficiency of the algorithm. The purpose of parallel algorithm is to make the algorithm can deal with the larger trajectory data set.

REFERENCES

[1] Z. Feng and Y. Zhu, "A survey on trajectory data mining: Techniques and applications," *IEEE Access*, vol. 4, pp. 2056–2067, 2016.

[2] J. Zhu, W. Jiang, A. Liu, G. Liu, and L. Zhao, "Effective and efficient trajectory outlier detection based on time-dependent popular route," *World Wide Web*, vol. 20, no. 1, pp. 111–134, Jan. 2017.

[3] H. Rai, M. H. Kolekar, N. Keshav, and J. Mukherjee, "Trajectory based unusual human movement identification for video surveillance system," in *Progress in Systems Engineering*. Las Vegas, NV, USA: Springer, 2015, pp. 789–794.

[4] Y. Cai, H. Jiang, H. Wang, and X. Chen, "Trajectory-based anomalous behaviour detection for intelligent traffic surveillance," *IET Intell. Transp. Syst.*, vol. 9, no. 8, pp. 810–816, Oct. 2015.

[5] P. Fu, H. Wang, K. Liu, X. Hu, and H. Zhang, "Finding abnormal vessel trajectories using feature learning," *IEEE Access*, vol. 5, pp. 7898–7909, 2017.

[6] X. Zhao, J. Zhang, and X. Qin, "*k* NN-DP: Handling data skewness in *kNN* joins using MapReduce," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 3, pp. 600–613, Mar. 2018.

[7] Y. Yang, J. Cai, H. Yang, J. Zhang, and X. Zhao, "TAD: A trajectory clustering algorithm based on spatial-temporal density analysis," *Expert Syst. Appl.*, vol. 139, Jan. 2020, Art. no. 112846.

[8] Y. Li, J. Cai, H. Yang, J. Zhang, and X. Zhao, "A novel algorithm for initial cluster center selection," *IEEE Access*, vol. 7, pp. 74683–74693, 2019.

[9] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: A partition-and-group framework," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*. New York, NY, USA: ACM, 2007, pp. 593–604.

[10] J.-G. Lee, J. Han, and X. Li, "Trajectory outlier detection: A partition-and-detect framework," in *Proc. IEEE 24th Int. Conf. Data Eng.*, Apr. 2008, pp. 140–149.

[11] N. R. Brisaboa, T. Gagie, A. Gómez-Brandón, G. Navarro, and J. R. Paramá, "Efficient compression and indexing of trajectories," in *Proc. Int. Symp. String Process. Inf. Retr.* Palermo, Italy: Springer, 2017, pp. 103–115.

[12] Y. Chen, P. Yuan, M. Qiu, and D. Pi, "An indoor trajectory frequent pattern mining algorithm based on vague grid sequence," *Expert Syst. Appl.*, vol. 118, pp. 614–624, Mar. 2019.

[13] C. Chen, D. Zhang, P. S. Castro, N. Li, S. Lin, S. Li, and Z. Wang, "iBOAT: Isolation-based online anomalous trajectory detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 806–818, 2nd Quart., 2013.

[14] J. Zhu, W. Jiang, A. Liu, G. Liu, and L. Zhao, "Time-dependent popular routes based trajectory outlier detection," in *Proc. Int. Conf. Web Inf. Syst. Eng.* Miami, FL, USA: Springer, 2015, pp. 16–30.

[15] J. Hua, Z. Yilong, and W. Xin, "Trajectory outlier detection based on trajectory information entropy distribution," *Appl. Res. Comput.*, vol. 35, no. 6, pp. 61–65, 2018.

[16] C. Yujun, P. Juhua, D. Jiahong, W. Yue, and X. Zhang, "Spatial–temporal traffic outlier detection by coupling road level of service," *IET Intell. Transp. Syst.*, vol. 13, no. 6, pp. 1016–1022, Jun. 2019.

[17] K. Zheng, G. Trajcevski, X. Zhou, and P. Scheuermann, "Probabilistic range queries for uncertain trajectories on road networks," in *Proc. 14th Int. Conf. Extending Database Technol.* New York, NY, USA: ACM, 2011, pp. 283–294.

[18] K. Zheng, Y. Zheng, X. Xie, and X. Zhou, "Reducing uncertainty of low-sampling-rate trajectories," in *Proc. IEEE 28th Int. Conf. Data Eng.*, Apr. 2012, pp. 1144–1155.

[19] J. Han, Z. Li, and L. A. Tang, "Mining moving object, trajectory and traffic data," in *Proc. Int. Conf. Database Syst. Adv. Appl.* Tsukuba, Japan: Springer, 2010, pp. 485–486.

[20] C. Qu, H. Yang, J. Cai, J. Zhang, and Y. Zhou, "DoPS: A double-peaked profiles search method based on the RS and SVM," *IEEE Access*, vol. 7, pp. 106139–106154, 2019.

[21] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: Algorithms and applications," *VLDB J. Int. J. Very Large Data Bases*, vol. 8, nos. 3–4, pp. 237–253, Feb. 2000.

[22] G. Yuan, S. Xia, L. Zhang, Y. Zhou, and C. Ji, "Trajectory outlier detection algorithm based on structural features," *J. Comput. Inf. Syst.*, vol. 7, no. 11, pp. 4137–4144, 2011.

[23] Z. Liu, D. Pi, and J. Jiang, "Density-based trajectory outlier detection algorithm," *J. Syst. Eng. Electron.*, vol. 24, no. 2, pp. 335–340, Apr. 2013.

[24] L. Liu, S. Qiao, Y. Zhang, and J. Hu, "An efficient outlying trajectories mining approach based on relative distance," *Int. J. Geograph. Inf. Sci.*, vol. 26, no. 10, pp. 1789–1810, Oct. 2012.

[25] X. Li, Z. Li, J. Han, and J.-G. Lee, "Temporal outlier detection in vehicle traffic data," in *Proc. IEEE 25th Int. Conf. Data Eng.*, Mar. 2009, pp. 1319–1322.

[26] R. Ramirez-Padron, D. Foregger, J. Manuel, M. Georgiopoulos, and B. Mederos, "Similarity kernels for nearest neighbor-based outlier detection," in *Proc. Int. Symp. Intell. Data Anal.* Tucson, AZ, USA: Springer, 2010, pp. 159–170.

[27] D.-G. Sim, O.-K. Kwon, and R.-H. Park, "Object matching algorithms using robust Hausdorff distance measures," *IEEE Trans. Image Process.*, vol. 8, no. 3, pp. 425–429, Mar. 1999.

[28] G. Yuan, S. Xia, L. Zhang, and Y. Zhou, "Structural outlier detection in trajectory database based on hierarchical tree," *INFORMATION*, vol. 15, no. 8, pp. 3595–3602, 2012.

[29] H. Parvin and B. Minaei-Bidgoli, "A clustering ensemble framework based on elite selection of weighted clusters," *Adv. Data Anal. Classif.*, vol. 7, no. 2, pp. 181–208, Jun. 2013.

[30] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Soft + hardwired attention: An LSTM framework for human trajectory prediction and abnormal event detection," *Neural Netw.*, vol. 108, pp. 466–478, Dec. 2018.

[31] R. Couillet and F. Benaych-Georges, "Kernel spectral clustering of large dimensional data," *Electron. J. Statist.*, vol. 10, no. 1, pp. 1393–1454, 2016.

[32] A. Nazari, A. Dehghan, S. Nejatian, V. Rezaie, and H. Parvin, "A comprehensive study of clustering ensemble weighting based on cluster quality and diversity," *Pattern Anal. Appl.*, vol. 22, no. 1, pp. 133–145, Feb. 2019.

[33] F. Rashidi, S. Nejatian, H. Parvin, and V. Rezaie, "Diversity based cluster weighting in cluster ensemble: An information theory approach," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 1341–1368, Aug. 2019.

[34] S. Liu and S. Wang, "Trajectory community discovery and recommendation by multi-source diffusion modeling," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 4, pp. 898–911, Apr. 2017.

[35] M. Cuturi, "Fast global alignment kernels," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 929–936.

[36] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jul. 2006, pp. 2169–2178.

XUJUN ZHAO received the M.S. degree in computer science and technology and the Ph.D. degree from the Taiyuan University of Technology. He is currently an Associate Professor with the School of Computer Science and Technology, TYUST. He is a member of China Computer Federation (CCF). His research interests include data mining and parallel computing.

YUANQI RAO was born in Shanxi, China, in 1995. He is currently pursuing the M.S. degree with the Department of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan, China. His current research interests include data mining and artificial intelligence.

JIANGHUI CAI is currently the Chief Professor of Computer Application Technology, Taiyuan University of Science and Technology, Taiyuan, China. He is the long-term member of the Institute for Intelligent Information and Data Mining. His research concerns the data mining and machine learning methods in specific backgrounds of astronomical informatics, seismology, and mechanical engineering. He is a Senior Member of China Computer Federation (CCF).

WENQIANG MA was born in Shandong, China, in 1993. He is currently pursuing the M.S. degree with the Department of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan, China. His current research interests include data mining and machine learning.