Notice of Retraction

“An Improved High-Density Sub Trajectory Clustering Algorithm”
by Xiaoming Liu, Luxi Dong, Chunlin Shang, Xiangda Wei
in IEEE Access, Volume 8, 2020

This paper has been found to be in violation of IEEE’s Publication Principles. IEEE hereby retracts the content of this paper.

It has been brought to the attention of the Editors that the submitting author, Luxi Dong, did not receive permission from his coauthors to submit the article to IEEE Access, or to include their names on the article. The submitting author also did not appear to own the dataset used in the article.

Due to the nature of this violation, reasonable effort should be made to remove all past references to this paper, and refrain from future references to this paper.

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 7, 2020, accepted February 8, 2020, date of publication February 14, 2020, date of current version March 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2974059

An Improved High-Density Sub Trajectory Clustering Algorithm

XIAOMING LIU, LUXI DONG, CHUNLIN SHANG, AND XIANGDA WEI
Beijing Key Laboratory of Urban Intelligent Traffic Control Technology, North China University of Technology, Beijing 100144, China

Corresponding author: Luxi Dong (281667195@qq.com)

This work is supported in part by the National Key Research and Development Program of China under Grant 2018YFB1601003, and in part by the Beijing Natural Science Foundation under Grant 8172018.

ABSTRACT TRACLUS algorithm based on partition-and-group framework could not be distinguished the optimal partitioning accurately when the migration of trajectory points on both sides of corridor middle line was greatly offset, and the algorithm was sensitive to the input parameters. According to above deficiency, an improved high-density sub-trajectory clustering algorithm (HTRACLUS_DL) is proposed under the practical application background of a traffic corridor identification. Initially, sub trajectories are divided based on the spatio-temporal characteristic similarity of trajectories. Furthermore, a sub-trajectory parallel boundary method is constructed, which has higher precision than the partitioning algorithm used in TRACLUS. Additionally, sub-trajectory clustering center neighborhoods possess local high density and surrounded by lower density sub trajectories. However, the different sub-trajectory clustering centers are heterogeneity. Finally, a new sub-trajectory clustering algorithm is robust to input parameters based on sub-trajectory entropy. Experimental results based on trajectory data of mobile phone user in two cities show that HTRACLUS_DL could be solved the deficiency of TRACLUS. At the same time, the method obtains better clustering result based on spatio-temporal characteristics of sub trajectory and does not depend on parameter selection. HTRACLUS_DL could be identified traffic corridor of urban group effectively.

INDEX TERMS Trajectory data, parallel boundary, high-density, sub trajectory clustering, entropy enter.

I. INTRODUCTION

The traffic corridor is the backbone of the complex urban traffic network. The traffic corridor carries the connection of the radiation regions in most areas. The unbalanced regional traffic demand leads to the change of the spatial trend of the traffic corridor, which affects the traffic benefit of the overall traffic network to a certain extent. With the development of Location Based Services and Cloud Storage technology, residents have accumulated massive heterogeneous trajectories [1]. The study of identifying urban traffic corridor based on trajectory data is helpful to understand the spatiotemporal characteristics of travel demand in specific regions and improve the spatial structure and the function of urban transportation.

In addition to describing the travel information of a single individual, the movement trajectory can also reflect the overall travel mode. Trajectory clustering analysis can effectively simplify the trajectory data and provide a computationally efficient method for identifying urban traffic corridor. Most current trajectory clustering analysis methods use the entire trajectory as the minimum unit for moving trajectory feature clustering. In 2007, Lee *et al.* [2] constructed a segmentation-clustering framework for moving trajectories, and proposed a new trajectory clustering algorithm based on the TRACLUS framework, the new algorithm reduces the clustering unit to a sub-trajectory and preserves the overall similarity of the trajectory to the greatest extent.

II. RELATED WORK

The related research work of trajectory clustering generally includes the following three aspects: trajectory model definition, trajectory similarity measurement and clustering methods.

In the definition of trajectory model, the trajectory data is a set of discrete point sequences arranged in ascending order of timestamps. Before mining the trajectory data, the trajectory model needs to be defined according to the application background, including geometric description [2]

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

and symbol description [3]. The geometric description of the moving trajectory represents discrete coordinate points with timestamps, which reflects the movement state of the trajectory. Its accuracy depends on the number of sampling points and the collection frequency. In practical applications, complex trajectories need to be simplified. The symbolic description of the moving trajectory represents a discrete sequence of region geometry. It has advantages in data storage and computational efficiency. It needs to be based on dividing the spatial region. Therefore, there is a problem with the granularity of the partition: too much granularity leads to the loss of common patterns of some moving trajectories, the granularity is too small to extract similar patterns.

Ning *et al.* [4] constructed a method of minimizing cost offloading, proposed a two-way matching algorithm to adjust frequency spectrum and meet user delay constraints, combined with deep reinforcement learning method to optimize system state and realize distributed computation. Ning *et al.* [5] obtained the closely related social characteristics of vehicles by constructing the triangle relation structure chart, estimated the node connection probability in the network model by combining the characteristics of vehicles and associated equipment, and established the CSPD algorithm based on convolution neural network. Ning *et al.* [6] built an intelligent system framework for vehicle edge computing based on deep reinforcement learning technology, established a communication and computing state model based on Finite Markov Chain, combined with two-way matching and deep reinforcement learning methods, jointly optimized task scheduling and network resource allocation strategies to maximize the quality of user experience (QoE). Ning *et al.* [7] constructed an energy-saving scheduling framework for MEC which enabled IOV to minimize the energy consumption of RSU under the constraint of task waiting time. Chen *et al.* [8] constructed an online track compression framework running in the mobile environment, which includes two stages: (1) online track mapping stage, a novel compressor based on the direction change of intersection, namely direction change compression (HCC), is designed to develop a lightweight but efficient map matcher; (2) track compression stage, based on spatial direction matching (SD-matching) can match the sparse GPS points to the road network and make full use of the vehicle GPS track data. Chen *et al.* [9] proposed an economic fast travel service. In the first stage, the offline historical taxi track data is mined to identify the shortest travel path based on the estimated travel time under any given starting and ending point. In the second stage, an online adaptive taxi dispatching algorithm is constructed to select the route and determine the optimal travel service according to the real-time request iterative calculation Service path. Chen *et al.* [10] proposed a two-stage probability framework called TripImputor, which is used to estimate the purpose of taxi travel and recommend services to passengers at the place where they get off. Ning *et al.* [11] built a three-layer VFC model to realize distributed traffic management and minimize the response time of vehicle collection and event release. Kong *et al.* [12] briefly introduced the latest technology and application of MCS in smart city.

In terms of trajectory similarity measurement, Vlachos *et al.* [13] used the Longest Common Subsequence (LCSS) similarity function to extract trajectories with the same characteristics. Guan *et al.* [14] divided multi-dimensional trajectories by calculating the spatial similarity (direction, speed, rotation angle, and position) of the trajectory segments. Han and Pan [15] constructed the time-space interaction method of the moving trajectory model, transformed the trajectory from the spatial topology of the road network to the Euclidean space. Jia *et al.* [16] calculated user hierarchical multi-granularity similarity under different weights based on the vector space model (VSM). Turchini *et al.* [17] detected, identified, calibrated and matched the similarity of user activity behaviors based on the trajectory chain grouping and cluster gaussian mixture clustering model. Cai *et al.* [18] used the improved OPTICS algorithm to extract the common behavior pattern of trajectories at the semantic level according to different application scenarios. Zafar *et al.* [19] mined frequent trajectory sets based on geographical location information and mobile users, constructed a mutual labeling model of trajectory spatial coordinates and traffic semantics to analyze trajectories similarity.

The trajectory clustering methods were divided four parts as follow. Firstly, the model-based clustering method was established a hidden Markov chain polynomial regression model and Bayesian model under constraints, and determined the membership of the trajectory clustering cluster model based on the maximum likelihood method, but the method is inefficient in modeling and calculating the fit [20], [21]. Secondly, the distance-based trajectory clustering method was proposed the trajectory clustering problem transformed into a feature vector solution and its aggregation problem, and a distance metric function and the corresponding aggregation Class method (K-Means, K-Medoids, EM algorithm, etc.) to achieve trajectory clustering [22], [23]. Thirdly, the grid-based clustering method such as SCI, MAFIA, ENCLUS are mentioned. All methods perform clustering by analyzing the optimal resolution of the grid [24]–[26]. Finally, the density-based clustering method, Chen *et al.* [27] proposed a DENTRAC algorithm based on a parameter less trajectory density function. Liao *et al.* [28] proposed a directed density fast clustering method (D-OPTICS) to extract the structural information of complex road networks. Yu *et al.* [29] proposed an enhanced trajectory model based on multi-feature trajectory similarity measure to better apply for traffic monitoring and traffic congestion prediction. Zhu *et al.* [30] created a recurrent convolutional neural network for modeling the complex nonlinear relationship among features, which could be used to predict road traffic conditions [31]. Wang *et al.* [32] obtained all the taxi trajectories crossing the same source-destination pairs, and then, the regular trajectories and anomalous trajectories were distinguished by applying adaptive hierarchical clustering based on an optimal number of clusters. Cai *et al.* [33] implemented a novel algorithm of hotspot similarity to improve the efficiency of summarizing the hotspot data, which was used to quantify the spatio-temporal distribution of the hotspots and determine the degree of attractiveness to the residents. Li *et al.* [34] proposed a new algorithm named Sparse learning based on clustering by fast search and find of density peaks (SL-CFSFDP). Compared to CFSFDP, the proposed algorithm can obtain d_c automatically and use sparse learning to determine the neighbors of each data point, removing irrelevant data points at the same time. Yan *et al.* [35] proposed a new clustering algorithm based on fitness proportionate sharing to map the problem into a multimodal optimization problem. The individuals with the highest density values were the cluster centers, the fitness proportionate sharing strategy was implemented in the identification results to overcome the sensitivity of uneven density values of cluster centers.

In summary, the existing trajectory clustering methods have a ‘‘segmentation-clustering’’ framework. However, the TRACLUS algorithm fails to comprehensively consider the similarity clustering of sub-trajectories. Different from all previous studies, three contributions of our study could be summarized as follows:

1) In the segmentation step, the judgment of similar trajectory conditions only considers the spatial distance but ignores the influence of temporal features on the similarity. When the trajectory points on both sides of the channel's central axis have a large offset, the MDL algorithm in the TRACLUS algorithm cannot accurately find the best partition point.

2) The spatial distance function used by the TRACLUS algorithm does not meet the triangular relationship, and traditional spatial indexing techniques cannot be directly applied. Therefore, a new spatial index needs to be built to reduce its complexity.

3) In the process of clustering, TRACLUS algorithm is sensitive to input parameters, and the slight fluctuation of parameters will lead to totally different clustering results.

In this paper, an improved high-density sub trajectory clustering algorithm (HTRACLUS_DL) is proposed to solve the problem of TRACLUS algorithm. In the trajectory segmentation step, two trajectories are divided based on the trajectory spatio-temporal feature similarity measure and the sub-trajectory parallel edge method. The rest of this paper is structured as follows. We summarize the related work of the trajectory clustering method in Section 2. Section 3 presents the improved high-density sub trajectory clustering algorithm in detail. Section 4 uses real trajectory data to verify the reliability of the algorithm. The conclusions are given in Section 5.

III. HTRACLUS_DL ALGORITHM

A. RELATED DEFINITIONS

The partition-group-based research framework is the basis for implementing most current trajectory clustering methods. In this paper, the ‘‘divide-cluster’’ idea is used to apply the entire trajectory. $Traj = \{Traj_1, \cdots, Traj_{num}\}$ is divided into several spatiotemporal sub-trajectory datasets. Firstly, the ordering sub-trajectory segments that change with time are obtained by using the partitioning algorithm proposed, and then the HTRACLUS_DL algorithm is used to cluster the sub-trajectory segments to obtain the sub-trajectory clustering set. The mathematical definitions of spatio-temporal trajectory data, feature point data, and sub-trajectory segment data are as follows:

Definition 1: Spatiotemporal trajectory data. The given spatiotemporal trajectory data *Traj* has a total of *num* trajectories, expressed as $Traj = \{Traj_1, \cdots, Traj_{num}\}$, each trajectory is composed of several multi-dimensional sample points, $\forall Traj_i = \{Q_1, \cdots, Q_{itotalnum}\}$, $1 \le i \le num$, *itotalnum* is the total number of sample points in the *i*-th spatio-temporal trajectory. Any sample point $Q_j = \left\{q_{jnum}, q_{trajid}, q_{lon}, q_{lat}, q_t\right\}$ in the spatiotemporal trajectory, expressed as the time q_t, the position of the sample point q_{jnum} in the trajectory q_{trajid} is (q_{lon}, q_{lat}), where $1 \le j \le itotalnum$.

Definition 2: Feature point data. The sub-trajectory of the *i*-th spatio-temporal trajectory is divided to obtain the feature point data representing the trajectory $chaTraj_i = \left\{c_{1,t}, \cdots, c_{ichanum,t}\right\}$, where *ichanum* represents the *i*-th spatio-temporal trajectory The total number of feature points, $c_{i,t}$ represents the feature points at the time *t* of the trajectory.

Definition 3: Sub-trajectory segment data. The feature point data of all spatio-temporal trajectories constitutes its sub-trajectory segment data. Assuming there are *l* sub-trajectory segments, then the sub-trajectory segment data is defined as $SubTraj_L = \{L_1, \cdots, L_k, \cdots, L_l\}$, where $L_k = \left\langle c_{k,t}, c_{k+1,t}\right\rangle (1 \le k \le l)$ are represented as adjacent feature points, L_k represents a directed sub-trajectory segment formed by two adjacent feature points.

B. TRAJECTORY SPATIO-TEMPORAL SIMILARITY MEASURE

As most current trajectory clustering algorithms fail to comprehensively consider the similarity measures of trajectory spatio-temporal features, this paper builds on the similarity of trajectory spatial distance, adds a similarity judgment method of temporal distance, and initially divides the trajectory set. The spatial similarity measurement of trajectories usually uses spatial distance as the measurement standard. The TRACLUS algorithm uses vertical distance, translation distance, and angular distance to measure the spatial similarity between trajectories, but it does not fully meet the measurement properties. In this paper, the spatial distance divided by trajectory is redefined, and three trajectory segments $L_i(s_i, e_i)$, $L_j(s_j, e_j)$ and $L_k(s_k, e_k)$ are assumed, where the midpoints of L_i, L_j and L_k are respectively m_i m_j and m_k, as shown in Figure 1.

Definition 4: Vertical distance. The definition of the vertical distance between L_i and L_j is shown in formula (1):

$$d_\perp(L_i, L_j) = \frac{l_{\perp ij1}^2 + l_{\perp ij2}^2}{l_{\perp ij1} + l_{\perp ij2}} \tag{1}$$

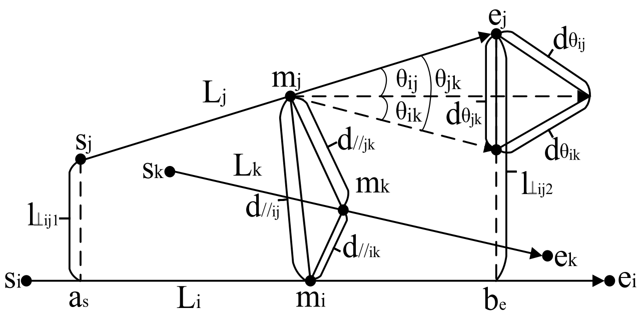


FIGURE 1. The spatial distance of trajectories diagram.

Definition 5: Translation distance. The translation distance between L_i and L_j is defined as the distance between m_i and m_j:

$$d_{//ij}(L_i, L_j) = \left\| m_i m_j \right\| \tag{2}$$

In the formula, $\left\| m_i m_j \right\|$ represents the Euclidean distance between m_i and m_j.

Definition 6: Angular distance. The angular distance between L_i and L_j is defined as the distance between the point e_i and e_j when L_i translates to the coincidence of m_i and m_j, as shown in formula (3):

$$d_{\theta_{ij}}(L_i, L_j) = \sqrt{\|m_i e_i\|^2 + \left\|m_j e_j\right\|^2 - 2\,\|m_i e_i\|\,\left\|m_j e_j\right\| \cos\theta_{ij}} \tag{3}$$

In the formula, θ_{ij} represents an included angle between L_i and $L_j (0 \leq \theta_{ij} \leq \pi)$.

In summary, the spatial distance between trajectories is composed of three parts: vertical distance, translation distance, and angular distance. The definition is shown in formula (4):

$$\begin{aligned} d(L_i, L_j) = \alpha_\perp * d_\perp(L_i, L_j) + \beta_{//} * d_{//ij}(L_i, L_j) \\ + \gamma_\theta * d_{\theta_{ij}}(L_i, L_j) \end{aligned} \tag{4}$$

In the formula, $\alpha_\perp$, $\beta_{//}$, and γ_θ have different values according to different application backgrounds. By default, except that $\alpha_\perp$ takes the value of 0, the others take the value of 1.

As it can be seen from the trajectory spatial distance definition diagram, the spatial distance takes into account factors such as the spatial position between trajectories, the trajectory length and its angle. Compared with the distance function in the TRACLUS algorithm, the spatial distance proposed in this paper also meets the metric property.

Theorem 1: The spatial distance between trajectories satisfies the metric property.

Proof: $d(L_i, L_j)$ completely satisfies non-negativity and symmetry, so it is only necessary to prove that $d(L_i, L_j)$ satisfies the metric property, the relationship of triangular inequality. As shown in Figure 1, for any trajectory segment L_i, L_j and L_k:

$$\begin{aligned} d(L_i, L_k) &= \beta_{//} * d_{//_{ik}}(L_i, L_k) + \gamma_\theta * d_{\theta_{ik}}(L_i, L_k) \\ &\leq \beta_{//} * [d_{//_{ij}}(L_i, L_j) + d_{//_{jk}}(L_j, L_k)] \\ &\quad + \gamma_\theta * \left[d_{\theta_{ij}}(L_i, L_j) + d_{\theta_{jk}}(L_j, L_k)\right] \end{aligned} \tag{5}$$

and:

$$\begin{aligned} &\beta_{//} * \left[d_{//_{ij}}(L_i, L_j) + d_{//_{jk}}(L_j, L_k)\right] \\ &\quad + \gamma_\theta * \left[d_{\theta_{ij}}(L_i, L_j) + d_{\theta_{jk}}(L_j, L_k)\right] \\ &= \left[\beta_{//} * d_{//_{ij}}(L_i, L_j) + \gamma_\theta * d_{\theta_{ij}}(L_i, L_j)\right] \\ &\quad + \left[\beta_{//} * d_{//_{jk}}(L_j, L_k) + \gamma_\theta * d_{\theta_{jk}}(L_j, L_k)\right] \\ &= d(L_i, L_j) + d(L_j, L_k) \end{aligned} \tag{6}$$

In summary, $d(L_i, L_k) \leq d(L_i, L_j) + d(L_j, L_k)$.

We have completed the proof. Therefore, the triangle inequality relationship is satisfied. Theorem 1 is to ensure that HTRACLUS_DL algorithm uses spatial indexing technology to achieve the purpose of reducing its complexity.

The time distance is calculated between adjacent spatiotemporal trajectories. The unit of measure is the direct factor that affects the sub-trajectory clustering results. References [36], [37] eliminate the unit dimension effect between spatiotemporal data through data preprocessing. Among them, reference [37] proposed a sliding window STS distance clustering algorithm with unequal long-term sequences. However, this method has high time complexity. The calculation efficiency is low.

This paper proposes a simplified pre-processing method for time distance. The GPS time of the trajectory is set as the time base point, and the time value is 0. Second is taken as the time unit, and the final time range is 0 to 864000. To facilitate the calculation of the time matrix, the current trajectory time was divided by the number of seconds in a day (864000), and the resulting value was used to construct a standardized linear time series. Assume that the longer trajectory time interval is $L_i(st_i, et_i)$, where st_i and et_i correspond to the starting time point and ending time point of L_i respectively, and the same can be shorter The trajectory time interval is $L_j(st_j, et_j)$, $\|L_i\| \geq \left\|L_j\right\|$. Comparing the two trajectory time intervals, the following three cases can be obtained, as shown in Figure 2. L_i contains L_j into two types: L_i completely contains and partially contains L_j; L_i does not contain L_j at all, where d_{L_j}, $d_{L_{j1}}$ and $d_{L_{j2}}$ respectively correspond to the time distance in the case of complete inclusion, partial inclusion, and no inclusion at all.

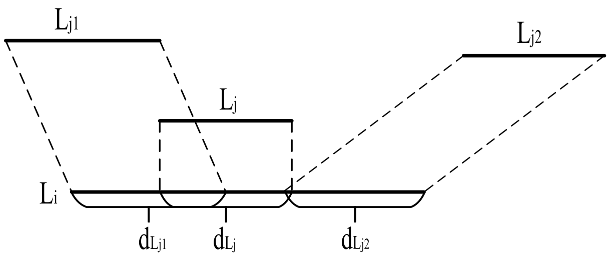


FIGURE 2. Three situations of spatio-temporal trajectory.

Through applying to actual data, there may be a large time interval span between two trajectories, and the nearest projection distance between two adjacent trajectories is taken as the time distance. Take the right projection as shown in Figure 3, project the starting time point st_{j1} of L_{j1} onto L_i and point st'_{i1}

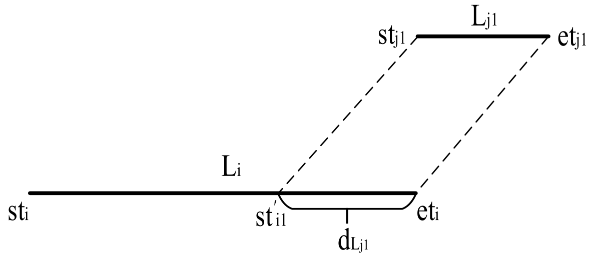


FIGURE 3. The right projection of trajectory.

from its et_i length $d_{L_{j1}}$, where $d_{L_{j1}} = |st'_{i1} - et_i|$. The time distance between the two trajectories L_i and L_j is defined as formula (7):

$$d_{time}(L_i, L_j) = \begin{cases} 0, & \text{if } L_i \subset L_j \\ \sqrt{\left|st'_{i1} - st_j\right|^2 + \left|et_i - et_j\right|^2}, & \text{if other} \end{cases} \tag{7}$$

In order to eliminate the problem of different unit dimensions of trajectory spatio-temporal distance, this paper USES standard score (z-score) method to standardize it, as shown in formula (8)~(10):

(1) Calculate the mean of absolute deviation:

$$\overline{S_{Traj}} = \frac{\left(\left|x_{1,Traj} - \overline{x}_{Traj}\right| + \cdots + \left|x_{n,Traj} - \overline{x}_{Traj}\right|\right)}{n} \tag{8}$$

where $x_{1,Traj}, \cdots, x_{n,Traj}$ are n metrics of $Traj$, $\overline{x}_{Traj}$ is the average of $Traj$:

$$\overline{x}_{Traj} = \frac{(x_{1,Traj} + \cdots + x_{n,Traj})}{n} \tag{9}$$

(2) Calculation of standard measurement values:

$$Z_{i,Traj} = \frac{(x_{i,Traj} - \overline{x}_{Traj})}{\overline{S_{Traj}}} \tag{10}$$

where $Z_{i,Traj}$ is the standardized metric value. Let the trajectory spatio-temporal distance after normalization be d'_{dist} and d'_{time}, respectively. Set the spatio-temporal distance weight $\omega\,(0 \leq \omega \leq 1)$ to adjust its sensitivity. The spatio-temporal distance between trajectory segments is defined as formula (11):

$$d_{ST} = \omega * d'_{dist} + (1 - \omega) * d'_{time} \tag{11}$$

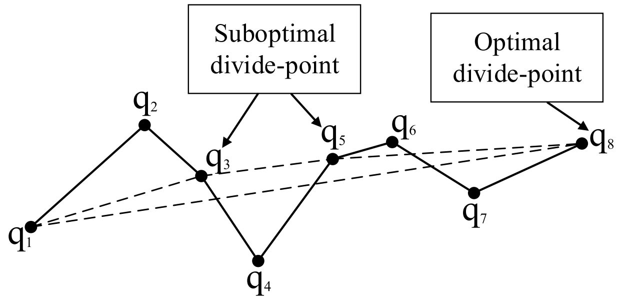


FIGURE 4. The failure schematic diagram of trajectory partition based on the MDL algorithm.

C. TRAJECTORY PARTITION METHOD BASED ON SUB-TRAJECTORY PARALLEL EDGES

The TRACLUS algorithm uses the MDL algorithm to divide the trajectory. The MDL algorithm uses the local optimal solution as the global optimal solution to improve the efficiency of the algorithm. However, the MDL algorithm in Figure 4 will fail, i.e. $MDL_p(q_1, q_8) \leq MDL_{notp}(q_1, q_8)$. The partition point in q_1q_8 trajectory segment should be after q_8, but the MDL algorithm q_3 and q_8 will be used as the dividing points because q_4 and q_6 satisfy $MDL_p(q_1, q_4) > MDL_{notp}(q_1, q_4)$ and $MDL_p(q_3, q_6) > MDL_{notp}(q_3, q_6)$, at this time the MDL algorithm cannot find the best dividing point.

In order to solve the shortcomings of the MDL partitioning algorithm, this paper proposes a partitioning method based on the parallel edges of the sub-trajectories, so that the divided sub-trajectories are surrounded by the parallel edges, and the forward detection of the sub-trajectory dividing points is realized. Construct the parallel edges with the detected trajectory points q_1 and q_3 in Figure 4, as shown in Figure 5, extract the trajectory points q_1, q_5 and q_8 in Figure 5 to find the first meeting the segmentation conditions of all feature points surrounded by the parallel edges. The previous trajectory point, and the next detected trajectory point is determined as the next trajectory point. If all points do not meet the dividing conditions (such as point q_8 in Figure 5), the next trajectory detected. Point is the next trajectory point of the last point in the sub-trajectory. The width threshold of the parallel edges is consistent with the length of the shortest sub-trajectory. The detailed description of the partition method based on the parallel edges of the sub-trajectories is shown in Figure 6.

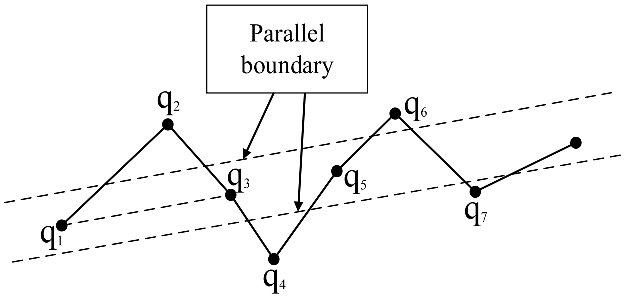


FIGURE 5. The schematic diagram of sub trajectory partition based on the parallel boundary method.

D. CLUSTERING ALGORITHM BASED ON HIGH-DENSITY SUB TRAJECTORY

The density-based clustering method segments the multidimensional trajectory set, and the number of initial clusters cannot be predicted, which results in the detection of irregular and noisy sub-trajectory segment shapes. In the DBSCAN algorithm, the trajectory points whose density is less than the density threshold are defined as noise, and the unrelated density regions are regarded as different clusters, but the disadvantage of this algorithm is that it is very sensitive to the threshold parameters (ε and *minlns*). In order to solve

Algorithm 1 Partitioning Method Based on Sub-Trace Parallel Edges

```
    Input: Any $Traj = \{q_1, \cdots, q_n\}$.
    Output: Sub-trajectory segment
    $SubTraj_L = \{L_1, \cdots, L_k, \cdots, L_l\}$
    Step1: $q_1$ is set as the starting point, and $q_2$ is set as the detection trajectory point.
    Step2: If the detection trajectory point is not $q_n$.
    Step3: Determine whether the current detection trajectory point is a partition point. If it is determined that the trajectory point is a partition point, construct a new sub-trajectory segment $SubTraj$, add it to $SubTraj_L$. Set the starting point to the current detection trajectory point, next to the trajectory point, return to Step2. If it is judged that it is not a partition point, go to the next step.
    Step4: Construct parallel edges according to the trajectory starting position and detection trajectory points.
    Step5: Find all points surrounded by parallel edges and record them as $T\_points$.
    Step6: Find the previous point of the first trajectory point that meets the partition conditions of $T\_points$, and modify the detection trajectory point to the next point, return to Step2.
    Step7: If $T\_points$ do not meet the partition conditions, the detection position is modified to the next trajectory point of the last point in $T\_points$, and return to Step2.
    Step8: Construct a new sub-trajectory segment $SubTraj$ according to the starting point and $q_n$, then add it to $SubTraj_L$.
    Step9: Return to $SubTraj_L$. Otherwise the algorithm ends.
```

the shortcomings of the DBSCAN algorithm, we propose an improved high-density sub-trajectory clustering algorithm based on clustering method of fast searching high-density points [38]–[40]. The main idea of this algorithm is that the local density between the cluster centers is higher than the density of the surrounding points. There is heterogeneity between different cluster centers. For any trajectory point, the algorithm only needs to calculate two important features: the local density ρ_i and its relative distance δ_i to the trajectory points with higher density depends on the distance d_{ij} between different trajectory points. The calculation formula for the local density ρ_i is shown in (12):

$$\rho_i = \sum_j X(d_{ij} - d_c) \tag{12}$$

In the formula, if $y < 0$, $X(y) = 1$, otherwise $X(y) = 0$. d_c represents the cutoff distance. δ_i represents the minimum distance of a point whose distance is greater than its local density, the calculation formula is shown in (13):

$$\delta_i = \min_{j:\rho_i > \rho_j} (d_{ij}) \tag{13}$$

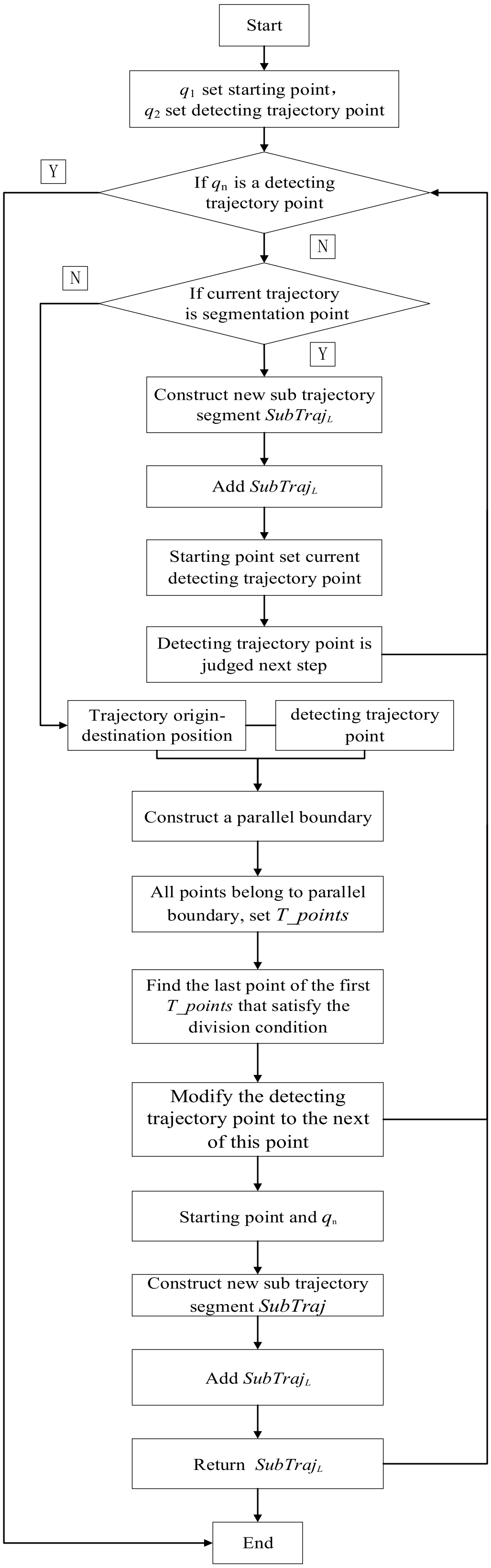

Start
q_1 set starting point，
q_2 set detecting trajectory point
Y
If q_n is a detecting trajectory point
N
N
If current trajectory is segmentation point
Y
Construct new sub trajectory segment $SubTraj_L$
Add $SubTraj_L$
Starting point set current detecting trajectory point
Detecting trajectory point is judged next step
Trajectory origin-destination position
detecting trajectory point
Construct a parallel boundary
All points belong to parallel boundary, set T_points
Find the last point of the first T_points that satisfy the division condition
Modify the detecting trajectory point to the next of this point
Starting point and q_n
Construct new sub trajectory segment $SubTraj$
Add $SubTraj_L$
Return $SubTraj_L$
End

FIGURE 6. The flowchart of sub trajectory parallel boundary method.

The maximum local density distance of the trajectory points is defined as $\delta_i = \max_j d_{ij}$. The cluster center is defined as the trajectory points with a larger local density and a longer distance. The remaining points will be clustered into the closest to center and greater than point clusters with their local density. In order to apply this clustering idea for the spatio-temporal sub-trajectory clustering algorithm, this article needs to modify some definitions:

Definition 7: The local density ρ_i of the sub-trajectory represents the number of other sub-trajectories within the cutoff distance. The calculation formula is shown in (14):

$$\rho_i = \sum\nolimits_{SubTraj_j \in (SubTraj_L - SubTraj_i)} \exp - \frac{\left(d_{ij}\right)^2}{d_c} \tag{14}$$

In the formula, the cutoff distance represented by d_c is completely different from the reference [40]. At the same time, the calculation result is a continuous value, which effectively avoids the same density data and facilitates subsequent processing.

Definition 8: The distance δ_i of the sub-trajectory segment represents the minimum distance to the trajectory segment. The distance of this sub- trajectory segment is higher than its local density. The calculation formula is shown in (15):

$$\delta_i = \begin{cases} \min\left\{d_{q_i,q_j}\right\}, & i \geq 2 \\ \max\left\{\delta_{q_j}\right\}, & i = 1 \end{cases} \tag{15}$$

In the formula, $\{q_i\}_{i=1}^n$ is a descending order combination of $\{p_i\}_{i=1}^n$, which satisfies the condition $\rho_{q_1} \geq \rho_{q_2} \geq \cdots \geq \rho_{q_n}$.

Definition 9: The sub-trajectory center clustering cluster $\{SC_i\}_{i=1}^m$ represents a set of sub-trajectory segments with larger ρ and δ, where m represents the number of sub-trajectory segments and the core sub-trajectory clustering cluster is generated.

Definition 10: The sub-trajectory noise represents a sub-trajectory segment having a larger ρ and a smaller δ. In this paper, the noise sub-trajectory is defined as the local density of the sub-trajectory is less than the average local density and the distance is greater than the average sub-trajectory distance.

In order to quickly select the sub-trajectory center clustering cluster $\{SC_i\}_{i=1}^m$, we define a new metric density-distance product.

Definition 11: The density-distance product (DD) represents the product of the density and distance of the sub-trajectory. Due to the large measurement scale of the local density and distance, the two are normalized. The calculation formula is shown in (16):

$$DD = k * \left(\frac{\rho_i}{\max(\rho)}\right) * \left(\frac{\delta_i}{\max(\delta)}\right) \tag{16}$$

In the formula, k is a constant.

The density-distance product comprehensively considers the distance between the local density and the sub-trajectory. The density-distance product takes the best value if and only if both take a larger value, that is, the larger the density-distance product is the sub-trajectory cluster center. Sort the density-distance product. According to the characteristics of the data distribution, the density-distance product value has a sudden change at a certain position. The cluster of the sub-trajectory center can be determined by observing the mutation point. The HTRACLUS_DL algorithm only needs to input one parameter, and the input parameters are less dependent on the attribute characteristics of the data itself. Theoretically, the input parameters have little effect on the segmentation results, and the sub-trajectory clustering algorithm is more robust to the input parameters.

The HTRACLUS_DL algorithm calculates the local density of each sub-trajectory $\{\rho_i\}_{i=1}^n$ after inputting a parameter to the sub-trajectory segment set. Secondly, sort the order of $\{\rho_i\}_{i=1}^n$, Calculate the distance $\{\delta_i\}_{i=1}^n$ for each sub-trajectory after sorting the local density values; then, calculate and sort $\{DD\}_{i=1}^n$ according to $\{\rho_i\}_{i=1}^n$ and $\{\delta_i\}_{i=1}^n$, determine the number of clusters, extract the central clusters. Finally, according to the distance between each sub-trajectory and each cluster center, the sub-trajectories will be clustered to different central clusters, the noise sub-trajectory will be deleted during the processing, the clustering ends, and the clustered sub-trajectory segment set is output. The specific algorithm flow is shown in Figure 7.

Algorithm 2 HTRACLUS_DL Algorithm

```
Input: any sub-trajectory segment $SubTraj_L = \{L_1, \cdots, L_k, \cdots, L_l\}$ and cutoff distance $d_c$.
Output: sub-trajectory clustering set $SubClu\_Traj_L = \{SubClu\_Traj_1, \cdots, SubClu\_Traj_{num}\}$.
Step1: Calculate $\{\rho_i\}_{i=1}^n$.
Step2: Sort $\{\rho_i\}_{i=1}^n$.
Step3: Calculate $\{\delta_i\}_{i=1}^n$.
Step4: Calculate $\{DD\}_{i=1}^n$ and sort it.
Step5: Determine the number of clusters $k$.
Step6: Extract the central clusters $\{SC_i\}_{i=1}^m$.
Step7: Any sub-trajectory segment $SubTraj_L$ is used to judge the noise sub-trajectory.
Step8: When the sub-trajectory segment $SubTraj_L$ is not noisy, then cluster it into the nearest central cluster with a higher local density.
Step9: Let $i = i + 1$, and return to Step5 for cyclic processing.
Step10: Otherwise, the algorithm ends.
```

E. PARAMETER SELECTION BASED ON ENTROPY THEORY

The HTRACLUS_DL algorithm only needs to input the parameter cutoff distance d_c, and then the value of d_c is selected by the entropy theory. When the clustering result is the worst, the number of other sub-trajectories in the sub-trajectory segment $SubTraj_L$ within the cutoff distance $\left|N_{d_c}(SubTraj_L)\right|$ is equal to the maximum entropy value. When the clustering result is better, the entropy value is the smallest. The value of d_c is selected based on the change trend of entropy with d_c, The definition of the sub-trajectory

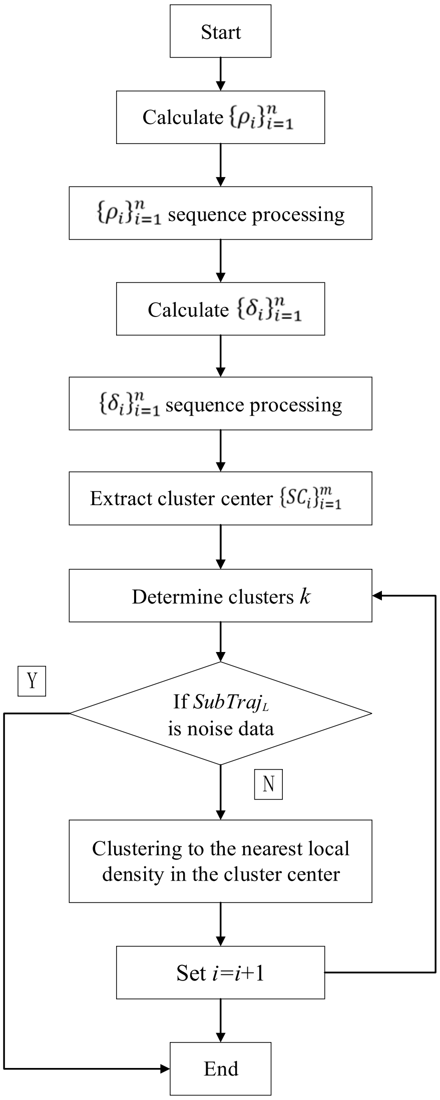

Start
Calculate $\{\rho_i\}_{i=1}^{n}$
$\{\rho_i\}_{i=1}^{n}$ sequence processing
Calculate $\{\delta_i\}_{i=1}^{n}$
$\{\delta_i\}_{i=1}^{n}$ sequence processing
Extract cluster center $\{SC_i\}_{i=1}^{m}$
Determine clusters k
Y
If $SubTraj_L$ is noise data
N
Clustering to the nearest local density in the cluster center
Set $i=i+1$
End

FIGURE 7. The flowchart of HTRACLUS_DL algorithm.

entropy is shown in formula (17):

$$Entropy = \sum_{i=1}^{N} p^D(L_i)\log_2 \frac{1}{p^D(L_i)} \tag{17}$$

In the formula, $p^D(L_i) = \frac{|N_{d_c}(L)|}{\sum_{j=1}^{n}|N_{d_c}(L)|}$, N represents the total number of sub-trajectory segments.

IV. CASE VERIFICATION

To verify the effectiveness of the HTRACLUS_DL algorithm, we use the real trajectory dataset of two urban mobile phone users. There are 14393 base stations in the dataset of city 1. The average coverage radiuses of base stations are 500 meters. When a mobile phone user received or sent a voice call or a message or an Internet service, the location register system is recorded data in base station. The average mobile phone records are produced by per person per day is 32.5. For privacy protection, personal information is not obtained from mobile phone data. Each user assigns with a matching number ID in dataset. Each record contains user ID, timestamp that a call was took, base station ID (set cell ID) and event type (voice call, message, surfing the Internet or passive communication), etc. We obtain the dataset of city 2. The average mobile phone records produced by per person per day is 38.3. There are 20493 base stations, the average coverage radiuses of base stations are 375 meters. The coverage radiuses of less than 500 meters is accounted for 90% of the total base stations. Due to a large demand for communications in the urban centers, the base stations of urban are dense, while the base stations of suburbs are relatively sparse.

TABLE 1. The dataset of city 1.

Link ID	Weight ω	Coordinates
10	0.9	LINESTRING(6332493.26886 2531529.89958, 6332689.36371 2527165.47534)
63	0.5	LINESTRING(6333669.83239 2524983.26267, 6333669.83239 2525181.64625)
108	0.3	LINESTRING(6334454.21178 2530537.98503, 6334454.21178 2531133.13354)
211	0.7	LINESTRING(6348180.78999 2512088.37353, 6348180.78999 2512286.75711)
……	……	……

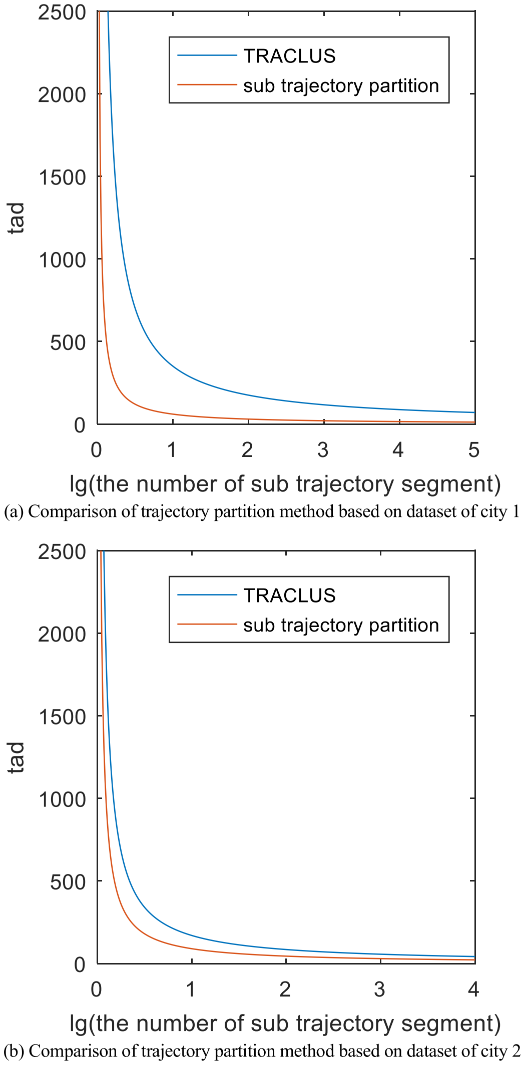
TABLE 2. The dataset of city 2.

Link ID	Weight ω	Coordinates
173	1.0	LINESTRING(6761900.0 3473400.0, 6761900.0 3473550.0)
406	0.4	LINESTRING(6758636.34695 3470099.44941, 6758640.89032 3470249.38059)
552	0.6	LINESTRING(6767900.0 3482200.0, 6767900.0 3482050.0)
497	0.3	LINESTRING(6761218.19805 3477318.19805, 6761324.26407 3477424.26407)
……	……	……

We count a total of 16,281,129 trajectories in city 1, and city 2 has 7,603,253 trajectories. Each processed dataset contains Link ID, Weight ω, Coordinates. For example, the datasets of city 1 and 2 are shown in Table 1 and 2 respectively.

A. EVALUATION METHOD OF PARALLEL EDGES OF SUB TRAJECTORIES

This paper improves the MDL segmentation algorithm in TRACLUS and proposes a sub-trajectory parallel edge partition method. The proposed algorithm and MDL algorithm are applied to the actual trajectory dataset, and then compared the accuracy and simplicity of that. The accuracy measure is defined as the total average distance (tad). The distance between each trajectory point and the corresponding sub



(a) Comparison of trajectory partition method based on dataset of city 1

(b) Comparison of trajectory partition method based on dataset of city 2

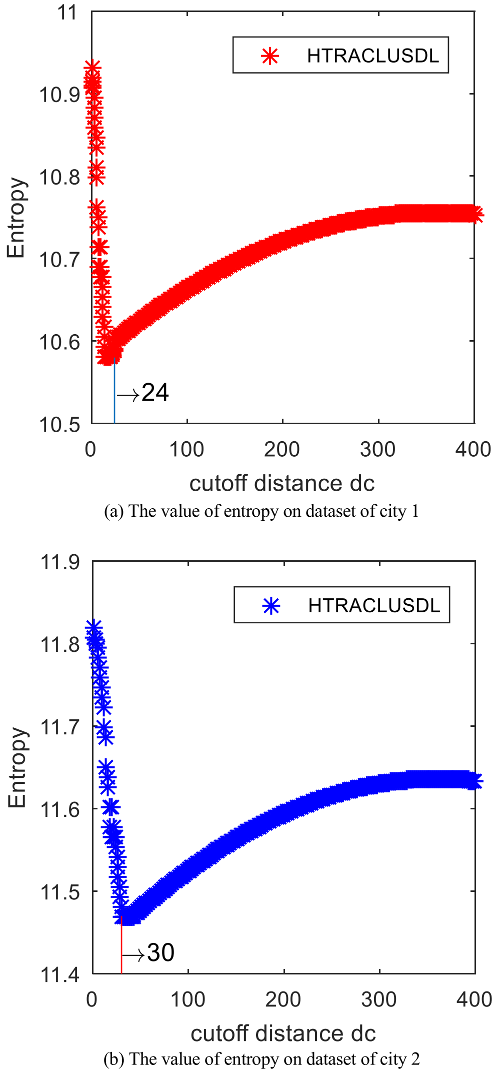
FIGURE 8. The comparison of TRACLUS algorithm and sub trajectory partition method.

trajectory is calculated to measure the mutual dissimilarity. The calculation formula is shown in formula (18):

$$tad = \frac{\sum_{i=1}^{n} \sum_{SubTraj_j \in Traj_i} \sum_{Q \in SubTraj_L} \lg \left[d\left(Q, SubTraj_L\right) \right]}{n} \tag{18}$$

In the formula, N represents the total number of sub-trajectory segments, $d\left(Q, SubTraj_L\right)$ represents the distance between the trajectory point Q and its sub-trajectory segment $SubTraj_L$.

The measure of simplicity is defined as the number of sub-trajectory segments. Select different sub-trajectory segment length parameters for verification, the larger the parameter, the longer the length of the sub-trajectory segments, and the total number of sub-trajectories will be smaller. Figure 8 shows under the same number of sub-trajectory segments, the *tad* value of the partition method is 300, which is smaller than the MDL algorithm, that is, the partition is more accurate. Under the same length parameter, the number of sub-trajectory segments divided by this paper is less than the MDL algorithm. Tables 3 and 4 show that the proposed method is more streamlined. In summary, the proposed method has better partition effect.



(a) The value of entropy on dataset of city 1

(b) The value of entropy on dataset of city 2

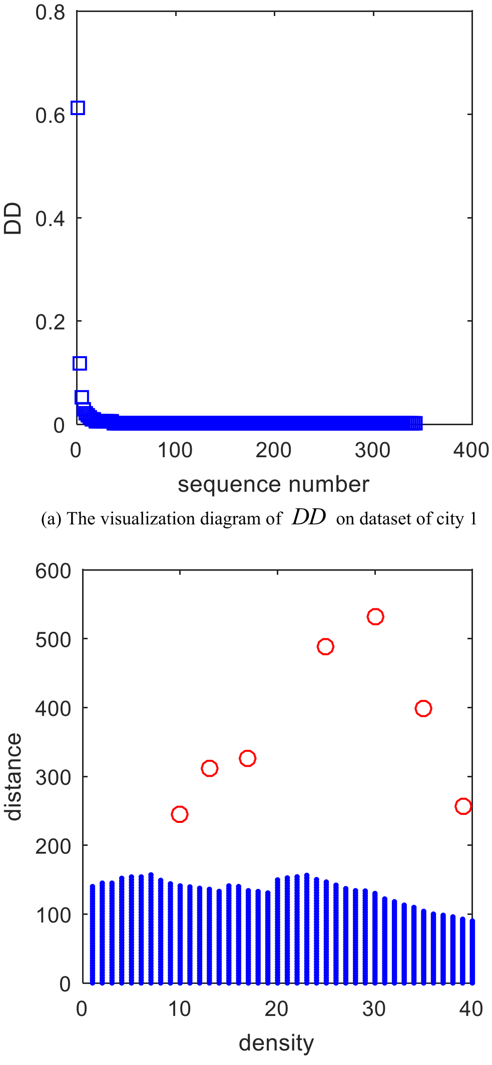
FIGURE 9. The entropy on two kinds of dataset.

TABLE 3. The results on sub-trajectory segment of city 1 based on the same distance.

Parameter	5	10	20	30	40
Sub-trajectory division method	116394	64106	33820	22061	11571
MDL algorithm	228940	112170	54120	32980	16800
Sub-trajectory variation	-49.1%	-42.8%	-37.5%	-33.1%	-31.1%

B. EVALUATION OF HIGH-DENSITY SUB-TRAJECTORY CLUSTERING ALGORITHM

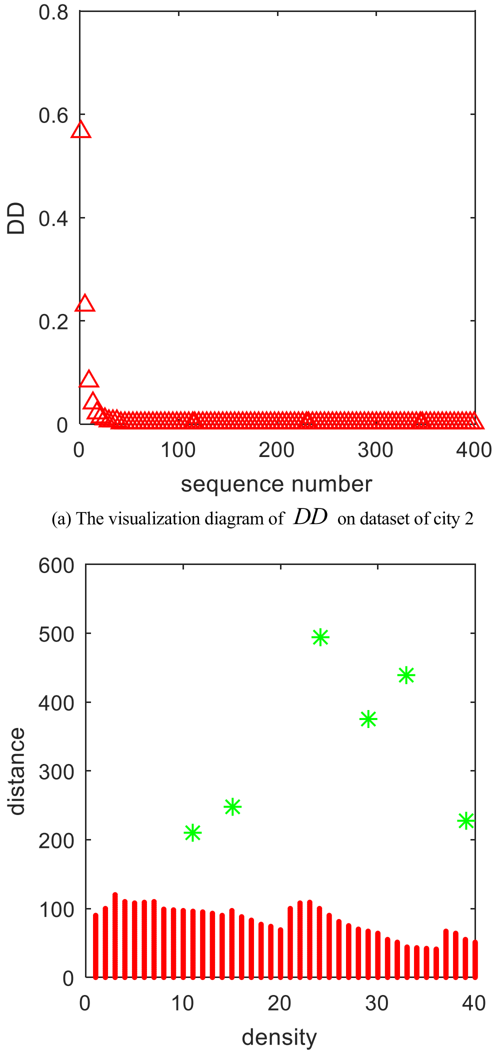
The HTRACLUS_DL algorithm requires only one parameter to determine the local density range. Figure 9 presents that the optimal cutoff distance for the city 1 trajectory dataset is 24,



(a) The visualization diagram of DD on dataset of city 1

(b) The coordinate graph of DD on dataset of city 1

FIGURE 10. The visualization and sequence diagram of *DD* on dataset of city 1.



(a) The visualization diagram of DD on dataset of city 2

(b) The coordinate graph of DD on dataset of city 2

FIGURE 11. The visualization and sequence diagram of on dataset of city 2.

TABLE 4. The results on sub-trajectory segment of city 2 based on the same distance.

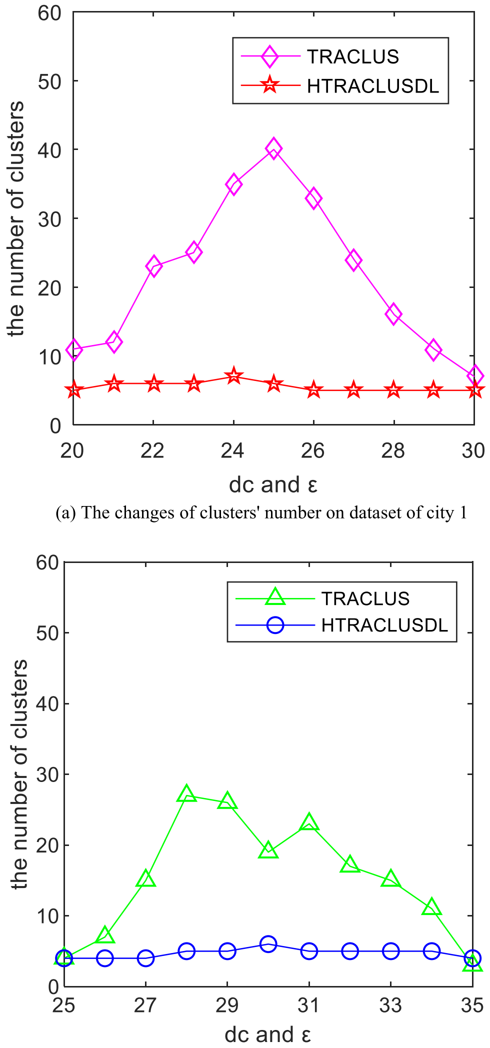
Parameter	5	10	20	30	40
Sub-trajectory division method	52426	52046	43910	22699	9368
MDL algorithm	125901	92509	71188	34143	13856
Sub-trajectory variation	-49.6%	-43.7%	-38.3%	-33.5%	-32.3%

and the optimal cutoff distance for the city 2 trajectory dataset is 30.

In the HTRACLUS_DL algorithm, the number of cluster centers is determined by the *DD* mutation points. The corresponding *DD* sorting and density-distance product visualization are shown in Figure 10. Figure 10 states there are 7 sub-trajectory segments with larger values of density and distance in the city 1 dataset, that is, there are 7 cluster centers. Figure 11 states there are a total of 6 cluster centers in the city 2 dataset.

The robustness of the HTRACLUS_DL algorithm is verified by the parameters. The sub-trajectory segments are clustered based on different cutoff distances. The value range of d_c in the city 1 trajectory dataset is [20], [30], and the value range of d_c in the city 2 trajectory dataset is [25], [35]. Comparing the trend of the number of clusters corresponding to different cutoff distance values, the results are shown in Figure 12. In the TRACLUS algorithm, using different neighborhood radius ε, the number of clusters changes greatly, while the number of clusters of the HTRACLUS_DL algorithm changes little. Therefore, the HTRACLUS_DL algorithm is more robust to input parameters.

The evaluation index of the clustering result is the sum square variance (*SSE*) and the noise penalty (*NP*). This index is first mentioned and used in reference [2]. We calculate *Q_Measure* to indicate the evaluation index, and the



(a) The changes of clusters' number on dataset of city 1

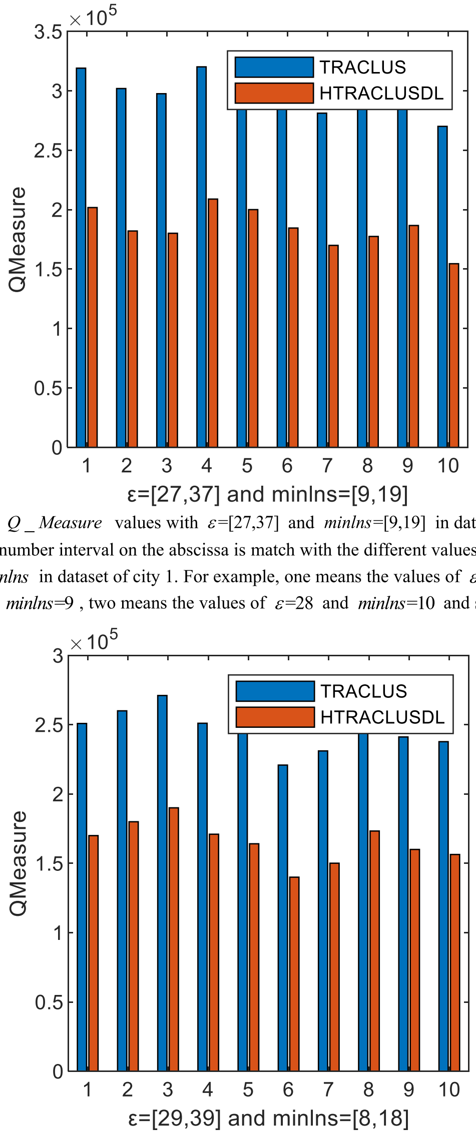
(b) The changes of clusters' number on dataset of city 2

FIGURE 12. The changes of different parameter diagram.

calculation formula is presented in (19):

$$Q\_Measure = SSE + NP = \sum_{K=1}^{M} \left(\frac{|c_K|}{2} * \sum\nolimits_{L_i \in c_K} \sum\nolimits_{L_j \in c_K} d\left(L_i, L_j\right)^2\right) + \frac{1}{2\,|N|} * \sum\nolimits_{\alpha \in N} \sum\nolimits_{\beta \in N} d\left(\alpha, \beta\right)^2 \quad (19)$$

In the formula, K represents the number of clusters, c_K represents the K-th cluster, N represents the number of noise sub-trajectories, and $d\left(L_i, L_j\right)$ represents the distance between two sub-trajectory segments. The smaller the $Q\_Measure$ value is, the better the clustering result will be. The parameter values of the TRACLUS algorithm provided ε and $minlns$ are completely different from the reference [2]. The parameter values are $\varepsilon = [27, 37]$ and $minlns = [9, 19]$ in the city 1 dataset, and the parameter values of city 2 dataset is $\varepsilon = [29, 39]$ and $minlns = [8, 18]$. Figure 13(a) illustrates that the average $Q\_Measure$ value of HTRACLUS_DL algorithm in the trajectory dataset of city 1 is 116,074.4 smaller than that of TRACLUS algorithm. In the trajectory dataset of city 2, Figure 13(b) illustrates the average $Q\_Measure$ value of HTRACLUS_DL algorithm is 80,865.9 smaller than TRACLUS algorithm, indicating that the clustering results of the proposed algorithm are better.



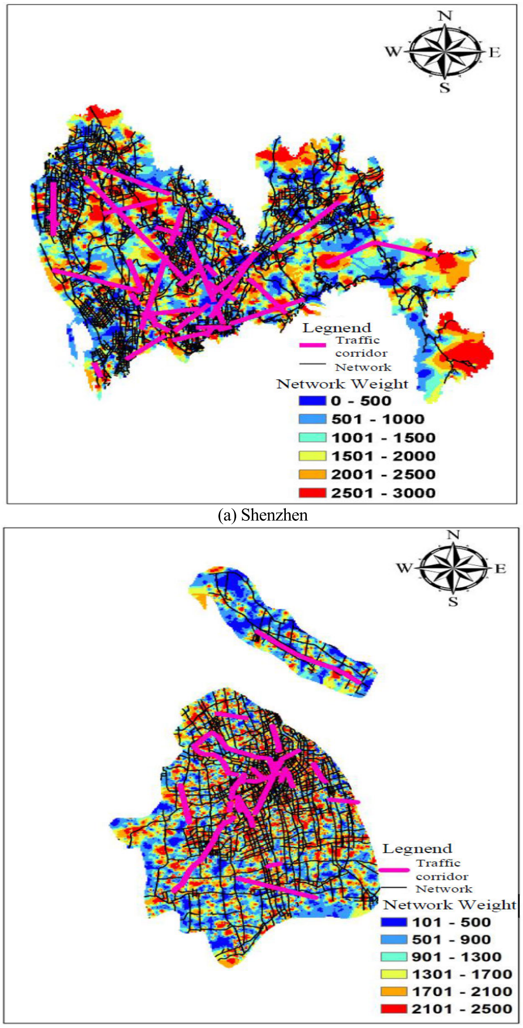
(a) The $Q\_Measure$ values with ε=[27,37] and $minlns$=[9,19] in dataset of city 1. The number interval on the abscissa is match with the different values of ε and $minlns$ in dataset of city 1. For example, one means the values of ε=27 and $minlns$=9 , two means the values of ε=28 and $minlns$=10 and so on.

(b) The $Q\_Measure$ values with ε=[29,39] and $minlns$=[8,18] in dataset of city 2. The number interval on the abscissa is match with the different values of ε and $minlns$ in dataset of city 2. For example, one means the values of ε=29 and $minlns$=8 , two means the values of ε=30 and $minlns$=9 and so on.

FIGURE 13. The comparison diagram of different trajectory data.

C. APPLICATION OF HIGH-DENSITY SUB-TRAJECTORY CLUSTERING ALGORITHM

Traffic corridors are mainly discrete Origin and Destination (OD) pairs distributed in various areas of the city, reflecting the spatial structure of the traffic in each area of the city. The structure of urban traffic corridors presents from the following three aspects.



(a) Shenzhen

(b) Shanghai

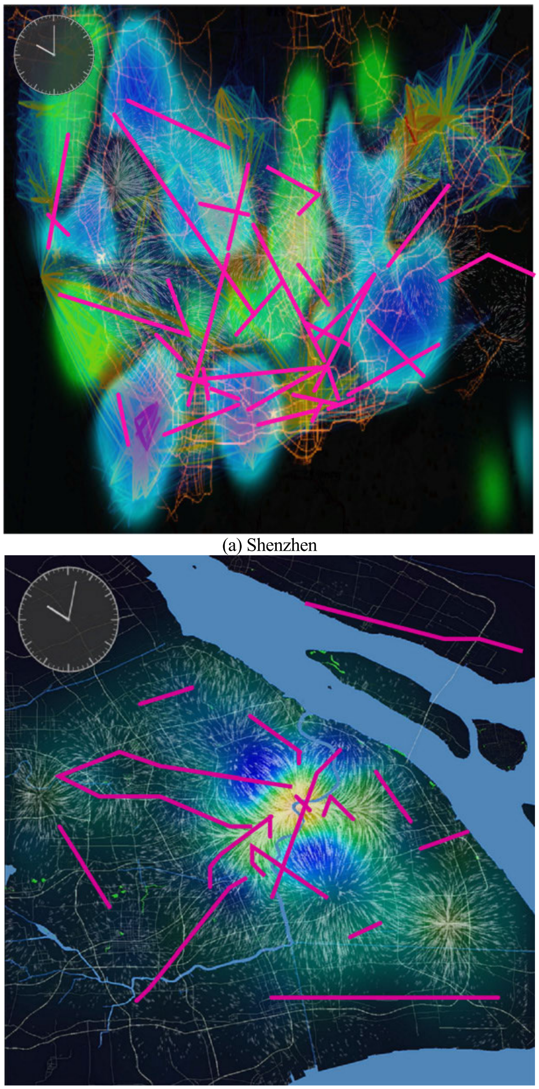
FIGURE 14. Comparison of urban traffic corridors and road networks.

1) MATCHING TRAFFIC CORRIDORS WITH ACTUAL ROAD NETWORK

Ideally, the traffic characteristic line should reflect the spatial structure of traffic between groups in different regions of the city. The identification results are compared with the actual road network. As shown in Figure 14, the base map is the daily average traffic flow weight density distribution of main lines in different cities. Figure 14(a) illustrates that the dense traffic network and the characteristic line in Shenzhen can well correspond to each other. Due to the complicated road network and the perfect planning of the surrounding land use, the travel mode is diversified, leading to scattering flow directions. On the contrary, some sections of the commuter flow corridor within the road outside urban (Longhua area near Shenzhen North Railway Station, Futian District and Luohu District) can correspond well.

2) COMPARISON OF URBAN TRAFFIC CORRIDORS AND PEOPLE TIDAL MOVEMENT

The base map is modified into a tidal motion map of the city during peak periods. It is assumed that the size of the urban population movement during the peak period represents the traffic flow within the traffic corridor. In the base map, red indicates the inflow of people, and blue presents the outflow of people. Figure 15 illustrates that the OD pairs of the traffic characteristic line in different cities can better connect the accumulation area and dissipation area of the flow.



(a) Shenzhen

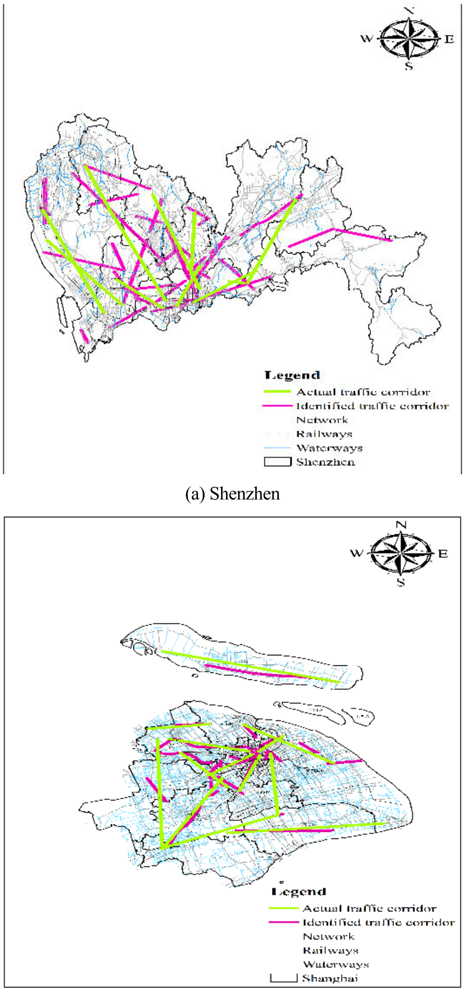
(b) Shanghai

FIGURE 15. Comparison of urban traffic corridors and tidal movement of people.

3) ACTUAL URBAN TRAFFIC CORRIDOR

By comparing the traffic corridor identification results with the actual corridors, Figure 16(a) shows Shenzhen's actual traffic corridors based on the HTRACLUS_DL algorithm:

1) Futian District and Nanshan District traffic corridor.
2) Futian District and Luohu District traffic corridor.
3) Futian District and Guangming District traffic corridor.
3) Baoan District and Nanshan District traffic corridor.
4) Longgang District and Luohu District traffic corridor.
5) Longhua District and Luohu District traffic corridor.
6) Pingshan District and Yantian District traffic corridor.



(a) Shenzhen

(b) Shanghai

FIGURE 16. **The traffic corridor identified with the actual corridor.**

7) Guanwai New District and urban central area traffic corridor.

Figure 16(b) shows the actual traffic corridors in Shanghai based on the HTRACLUS_DL algorithm:

1) Songjiang District, Jiading District, and the urban center district traffic corridor.

2) Hongkou District and the urban center district traffic corridor.

3) Pudong New District and the urban center district traffic corridor.

4) Fengxian District and Songjiang District Traffic corridor.

5) Chongming Island traffic corridor.

6) Outer ring road traffic corridor.

Figure 16 shows that the proposed method can better match the traffic corridor identification results with the actual corridors. There are 7 major traffic corridors in Shenzhen and 6 main corridors in Shanghai respectively.

V. CONCLUSION

With the accumulation of the trajectory, it is particularly important to mine the intrinsic meaning of the trajectory data. Trajectory clustering is one of the methods to effectively analyze the trajectory data and its characteristics. The TRACLUS algorithm is widely used in the field of trajectory clustering, but there are still two shortcomings, on the one hand, when the trajectory points on both sides of the central axis of the traffic corridor are deviated greatly in the trajectory partition, it is impossible to find the optimal point accurately; on the other hand, the algorithm is highly sensitive to input parameters. In view of the above-mentioned shortcomings, we propose an improved high-density sub-trajectory clustering algorithm (HTRACLUS_DL). This algorithm constructs a partition method based on the parallel edges of the sub-trajectories in the trajectory division process. The HTRACLUS_DL algorithm is used in the sub-trajectory clustering, calculating sub-trajectory entropy will enhance the robustness of the algorithm to input parameters. The verification shows that the HTRACLUS_DL algorithm can better solve the disadvantages of TRACLUS and achieve better clustering results. The HTRACLUS_DL algorithm can be widely used in the fields of traffic corridors identification and improvement of urban spatial structure and functions.

REFERENCES

[1] Y. Liu, C. Kang, and F. Wang, "Towards big data-driven human mobility patterns and models," *Geomatics Inf. Sci. Wuhan Univers*, vol. 39, no. 6, pp. 660–666, 2014.

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

[3] S. Dodge, R. Weibel, and E. Forootan, "Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects," *Comput., Environ. Urban Syst.*, vol. 33, no. 6, pp. 419–434, Nov. 2009.

[4] Z. Ning, P. Dong, M. S. Obaidat, X. Hu, L. Guo, Y. Guo, J. Huang, B. Hu, Y. Li, and X. Wang, "When deep reinforcement learning meets 5G-enabled vehicular networks: A distributed offloading framework for traffic big data," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1352–1361, Feb. 2020.

[5] Z. Ning, Y. Feng, M. Collotta, X. Kong, X. Wang, L. Guo, X. Hu, and B. Hu, "Deep learning in edge of vehicles: Exploring trirelationship for data transmission," *IEEE Trans Ind. Informat.*, vol. 15, no. 10, pp. 5737–5746, Oct. 2019.

[6] Z. Ning, P. Dong, F. Xia, J. J. Rodrigues, and X. Wang, "Deep reinforcement learning for vehicular edge computing: An intelligent offloading system," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 10, no. 6, p. 60, 2019.

[7] Z. Ning, J. Huang, X. Wang, J. J. P. C. Rodrigues, and L. Guo, "Mobile edge computing-enabled Internet of Vehicles: Toward energy-efficient scheduling," *IEEE Netw.*, vol. 33, no. 5, pp. 198–205, Sep. 2019.

[8] C. Chen, Y. Ding, X. Xie, S. Zhang, Z. Wang, and L. Feng, "TrajCompressor: An online map-matching-based trajectory compression framework leveraging vehicle heading direction and change," *IEEE Trans. Intell. Transp. Syst.*, to be published.

[9] C. Chen, D. Zhang, X. Ma, B. Guo, L. Wang, Y. Wang, and E. Sha, "Crowddeliver: Planning city-wide package delivery paths leveraging the crowd of taxis," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1478–1496, Jun. 2017.

[10] C. Chen, S. Jiao, S. Zhang, W. Liu, L. Feng, and Y. Wang, "TripImputor: Real-time imputing taxi trip purpose leveraging multi-sourced urban data," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 10, pp. 3292–3304, Oct. 2018.

[11] Z. Ning, J. Huang, and X. Wang, "Vehicular fog computing: Enabling real-time traffic management for smart cities," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 87–93, Feb. 2019.

[12] X. Kong, X. Liu, M. Li, L. Wan, F. Xia, and B. Jedari, "Mobile crowdsourcing in smart cities: Technologies, applications, and future challenges," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8095–8113, Oct. 2019.

[13] M. Vlachos, D. Gunopoulos, and G. Kollios, "Discovering similar multidimensional trajectories," in *Proc. 18th Int. Conf. Data Eng.*, Feb./Mar. 2002, pp. 673–684.

[14] Y. Guan, X. Shi-Xiong, Z. Yong, and Z. Lei, "Trajectory clustering algorithm based on structural similarity," *J. Commun.*, vol. 32, no. 9, pp. 103–110, 2011.

[15] H. Zhao, Q. Han, H. Pan, and G. Yin, "Spatio-temporal similarity measure for trajectories on road network," in *Proc. 4th Int. Conf. Internet Comput. Sci. Eng.*, Dec. 2009, pp. 189–193.

[16] R. R. Jia, S. G. Liu, and Q. L. Sun, "User similarity analysis based on location trajectory data," *Comput. Digit. Eng.*, vol. 44, no. 8, pp. 1523–1527, 2016.

[17] F. Turchini, L. Seidenari, and A. Del Bimbo, "Understanding and localizing activities from correspondences of clustered trajectories," *Comput. Vis. Image Understand.*, vol. 159, pp. 128–142, Jun. 2017.

[18] G. Cai, K. Lee, and I. Lee, "Mining mobility patterns from geotagged photos through semantic trajectory clustering," *Cybern. Syst.*, vol. 49, no. 4, pp. 234–256, 2018.

[19] A. Zafar, M. Kamran, S. A. Shad, and W. Nisar, "A robust missing data-recovering technique for mobility data mining," *Appl. Artif. Intell.*, vol. 31, nos. 5–6, pp. 425–438, Jul. 2017.

[20] S. Ghassempour, F. Girosi, and A. Maeder, "Clustering multivariate time series using hidden Markov models," *Int. J. Environ. Res. Public Health*, vol. 11, no. 3, pp. 2741–2763, 2014.

[21] M. Lázaro-Gredilla and S. A. Van Vaerenbergh, "Gaussian process model for data association and a semidefinite programming solution," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 11, pp. 1967–1979, Nov. 2014.

[22] N. Pelekis, I. Kopanakis, G. Marketos, I. Ntoutsi, G. Andrienko, and Y. Theodoridis, "Similarity search in trajectory databases," in *Proc. 14th Int. Symp. Temporal Represent. Reasoning (TIME)*, 2007, pp. 129–140.

[23] I. Sanchez, Z. M. M. Aye, B. I. P. Rubinstein, and K. Ramamohanarao, "Fast trajectory clustering using hashing methods," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 3689–3696.

[24] H. Zhao, X. Y. Liu, and H. Q. Cui, "Grid-based clustering algorithm," *Comput. Technol. Develop.*, vol. 20, no. 9, pp. 83–85, 2010.

[25] J. Wu, Y. Zhu, L. Wang, and T. Ku, "Hot routes detection algorithm based on grid clustering," *J. Jilin Univ. (Eng. Technol. Ed.)*, vol. 45, no. 1, pp. 274–282, 2015.

[26] Z. Y. Ni, "Trajectory pattern mining and route planning," M.S. thesis, Dept. Comput. Technol., Univ. Electron. Sci. Technol. China, Chengdu, China, 2017.

[27] C.-S. Chen, C. F. Eick, and N. J. Rizk, "Mining spatial trajectories using non-parametric density functions," in *Proc. Int. Workshop Mach. Learn. Data Mining Pattern Recognit.* Berlin, Germany: Springer, 2011, pp. 496–510.

[28] L. Liao, X. Jiang, L. Luming, H. Lai, and F. Zou, "A fast method of FCD trajectory data clustering based on the directed density," *J. Geo Inf. Sci.*, vol. 17, no. 10, pp. 1152–1161, 2015.

[29] Q. Yu, Y. Luo, C. Chen, and S. Chen, "Trajectory similarity clustering based on multi-feature distance measurement," *Int. J. Speech Technol.*, vol. 49, no. 6, pp. 2315–2338, Jun. 2019.

[30] J. Zhu, C. Huang, M. Yang, and G. P. Cheong Fung, "Context-based prediction for road traffic state using trajectory pattern mining and recurrent convolutional neural networks," *Inf. Sci.*, vol. 473, pp. 190–201, Jan. 2019.

[31] A. Agudo and F. Moreno-Noguer, "Robust spatio-temporal clustering and reconstruction of multiple deformable bodies," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 971–984, Apr. 2019.

[32] Y. Wang, K. Qin, P. Zhao, and Y. Chen, "Detecting anomalous trajectories and behavior patterns using hierarchical clustering from taxi GPS data," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 1, p. 25, 2018.

[33] L. Cai, F. Jiang, W. Zhou, and K. Li, "Design and application of an attractiveness index for urban hotspots based on GPS trajectory data," *IEEE Access*, vol. 6, pp. 55976–55985, 2018.

[34] P. Li, X. Deng, L. Zhang, J. Gan, J. Li, and Y. Li, "Sparse learning based on clustering by fast search and find of density peaks," *Multimedia Tools Appl.*, vol. 78, no. 23, pp. 33261–33277, Dec. 2019.

[35] X. Yan, M. Razeghi-Jahromi, A. Homaifar, B. A. Erol, A. Girma, and E. Tunstel, "A novel streaming data clustering algorithm based on fitness proportionate sharing," *IEEE Access*, vol. 7, pp. 184985–185000, 2019.

[36] L. Qin, W. Kaile, and R. Weixiong, "Non-equal time series clustering algorithm with sliding window STS distance," *J. Frontiers Comput. Sci. Technol.*, vol. 9, no. 11, pp. 1301–1313, 2015.

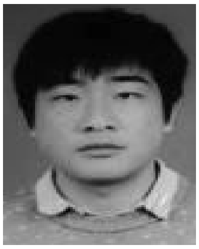
[37] L. Wang, J. Meng, K. X. Peng, and P. Xu, "Similarity dynamical clustering algorithm based on multidimensional shape features for time series," *Chin. J. Eng.*, vol. 39, no. 7, pp. 1114–1122, 2017.

[38] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.

[39] K. Sun, X. Geng, and L. Ji, "Exemplar component analysis: A fast band selection method for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 5, pp. 998–1002, May 2015.

[40] Y.-W. Chen, D.-H. Lai, H. Qi, J.-L. Wang, and J.-X. Du, "A new method to estimate ages of facial image for large database," *Multimedia Tools Appl.*, vol. 75, no. 5, pp. 2877–2895, Mar. 2016.

XIAOMING LIU received the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, in 2004. He is currently a Professor with the College of Electrical and Control Engineering, North China University of Technology. His research interests include traffic flow theory and intelligent traffic control.



LUXI DONG was born in 1993. He is currently pursuing the Ph.D. degree. His research interests include control science and engineering, intelligent traffic control, and data mining.

CHUNLIN SHANG was born in 1989. He is currently pursuing the Ph.D. degree. His research interests include control science and engineering and intelligent traffic control.

XIANGDA WEI was born in 1993. He is currently pursuing the master's degree. His research interests include control science and engineering and intelligent traffic control.