An Adaptive Staying Point Recognition Algorithm Based on Spatiotemporal Characteristics Using Cellular Signaling Data

Ming Cai, Zixuan Zhang, Chen Xiong, and Chao Gou

***Abstract*— Cellular signaling data (CSD) have attracted unprecedented attention due to their large size, long observation period, and high followability. Before applying CSD, a series of data processing steps are indispensable; among those steps, staying point recognition is the basis for recognizing individual travel states and thus the influence of further application of CSD. Previous work indicates that the existing staying point recognition algorithms have two common aspects. One is the requirement of a fixed spatiotemporal threshold to analyze the user's travel characteristics. The other is the insufficiency of accuracy assessment, which indicates that further studies are expected owing to the lack of ground truth data in CSD. In this work, a "spatiotemporal window"-based algorithm is proposed to recognize individual staying and moving states. First, an iterative-learning-based model is designed to cluster individual trajectory points without predefined spatiotemporal thresholds. Then, rules to distinguish the staying or moving cluster are made from individual travel characteristics. Moreover, verification work is carried out by collecting volunteers' ground truth data using our developed smartphone application, which achieves an accuracy of 91.3%. Finally, the results demonstrate the effectiveness and robustness of the algorithm through the performance of comparison and sensitivity analyses.**

***Index Terms*— Cellular signaling data, staying point recognition, spatiotemporal window, clustering method.**

I. Introduction

BASED on the different generation mechanisms of mobile phone data, Calabrese *et al.* [1] categorized the data into event-driven and network-driven types. Event-driven data include call detail records (CDRs) and Internet protocol detail records (IPDRs), which are mainly used for billing. Cellular signaling data (CSD) are a network-driven type of data that have two main generation mechanisms. First, CSD are generated in backend servers when communication occurs between a cell phone and a base station, such as switching on/off the phone, making/receiving calls, periodically updating location, updating location area, and sending or receiving messages. Second, based on communication-related work [2], it was found that there was a heartbeat mechanism in CSD, i.e., a smart cell phone still sends a handshake signal to the nearest base station, even if a person does not use the cell phone. The generation frequency of the heartbeat data in the 2G/3G communication mode is 30 minutes, and in the 4G mode is 5–10 minutes, depending on the rules of communications corporations.

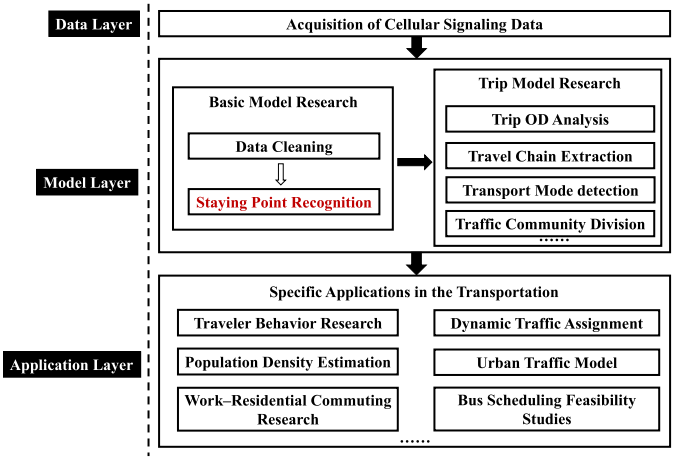

Fig. 1. Application framework of CSD in the transportation.

With the development of technology, people are becoming more reliant on cell phones, which generate abundant data describing user trajectories. CSD are characterized by large coverage areas, short sampling periods, long observation periods, and strong followability. It was previously found that the primary process of a CSD application contains three layers: a data layer, a model layer, and an application layer, as shown in Fig.1.

The data layer and model layer pose difficulties that directly influence the effectiveness of CSD applications. The model layer consists of a basic model and a trip model. Based on the staying point recognition result, the individual travel spatiotemporal characteristics [3] can be extracted, the mobility patterns [4], [5] can be derived, the origin-destination trip matrix [6] can be calculated, and the individual trip chains [7], [8] can be obtained. Furthermore, based on the trip trains, individual transport modes [9], [10] can be detected,

Manuscript received 31 July 2020; revised 16 March 2021 and 14 May 2021; accepted 30 June 2021. Date of publication 2 August 2021; date of current version 9 August 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 12002403 and Grant U1811463 and in part by the Fundamental Research Funds for the Central Universities, Sun Yat-sen University under Grant 2021qntd08. The Associate Editor for this article was L. Li. *(Corresponding author: Chen Xiong.)*

The authors are with the School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510006, China (e-mail: caiming@mail.sysu.edu.cn; zhangzx65@mail2.sysu.edu.cn; xiongch8@mail.sysu.edu.cn; gouchao@mail.sysu.edu.cn).

Digital Object Identifier 10.1109/TITS.2021.3094636

travel flow characteristics [11], [12] can be analyzed, and traveler behavior [13] can be researched. Overall, staying point recognition is an important prerequisite for transforming CSD trajectories into traffic semantics, where CSD show great potential application value. Besides, CSD are also widely used in dynamic traffic assignment [14], [15], route choice estimation [16], urban traffic model construction [17], work–residential commuting research [18], [19], and bus scheduling feasibility studies [20], [21]. Therefore, among these models, staying point recognition is a crucial module, and the accuracy of the algorithm directly influences the effectiveness of other models and the studies of CSD in transportation.

At present, the staying point is widely regarded as the starting and ending points of residents' trips. The CSD travel trajectory of each user consists of a sequence of spatiotemporal points that can be described as $\mathcal{PP} = \left\{ \begin{matrix} P_1\left(lng_1, lat_1, time_1\right), P_2\left(lng_2, lat_2, time_2\right), \ldots, \\ P_n\left(lng_n, lat_n, time_n\right) \end{matrix} \right\}$, and each point has its own attribute. The purpose of staying point recognition is to calculate the points attributes and distinguish whether the points belong to staying or moving points. Then those points could be described as $\mathcal{PP}' = \left\{ \begin{matrix} P_1(lng_1, lat_1, time_1, label_1), P_2(lng_2, lat_2, time_2, \\ label_2), \ldots, P_n(lng_n, lat_n, time_n, label_n) \end{matrix} \right\}$, where *label* means staying or moving state. Through staying point recognition, the spatiotemporal trajectory can be converted into traffic semantic information.

However, the existing staying recognition algorithms have two main limitations. First, these algorithms both require fixed empirical thresholds and numbers of clusters. Second, it is rarely mentioned that the algorithms are assessed by ground truth data.

This study proposes a novel algorithm based on spatiotemporal independence to address CSD staying point recognition and threshold setting and accuracy assessment issues. The layered cleaning algorithm is first used to clean the CSD to address the data uncertainties and redundancy properly. Then, a spatiotemporal window-based algorithm is proposed to cluster individual trajectory points without a predefined spatiotemporal threshold. A definition is established to judge each cluster's attribute, assigning all the basic clusters as staying or moving. Finally, a merging operation is carried out to combine adjacent clusters with the same attribute together, and thus, an individual's one-day staying points are obtained. In contrast to existing works, the main contributions of this work can be summarized as follows:

1. A novel spatiotemporal window-based algorithm is proposed based on the characteristics of individual travel trajectories, which can accurately identify the staying points of CSD. This method exhibits strong adaptability and low computational complexity, and it does not require the number of clusters or other spatiotemporal thresholds to be set.
2. An iterative-learning-based model is designed to cluster spatiotemporal trajectory points without predefined spatiotemporal thresholds. Through experimental verification, the method can be well applied to CSD and GPS data.
3. This is the first study in which verification work for cellular signaling data staying point recognition is performed by using our developed smartphone application.

The remainder of the paper is organized as follows. Section 2 is the literature review of the methods and algorithms of staying point recognition; Section 3 presents the algorithm of CSD staying point recognition; Section 4 presents the accuracy assessment and factor analysis of the proposed method; Section 5 outlines additional research; and Section 6 is the conclusion.

II. Literature Review

According to previous studies, there are three kinds of staying point recognition methods, including spatiotemporal rule-based, clustering-based, and machine learning-based methods.

A. Methods Based on Spatiotemporal Rules

In this kind of method, trajectory points are considered staying points when the duration of stay and the range of activity both satisfy the conditions determined by the spatial and temporal thresholds. Generally, these methods choose the coverage of the base station as the spatial threshold, which is usually 500 meters [23], and choose temporal threshold based on experiences, which is usually 30 minutes [22], [23] or 15 minutes [24]. This kind of method is simple with low computational complexity, but it is not interpretable and universally applicable to choose spatiotemporal thresholds based on the experiences and the nature of data.

B. Methods Based on Clustering

Staying point recognition of spatiotemporal CSD trajectories can be considered a case of clustering analysis. The clustering algorithms used in CSD are mainly based on K-means and DBSCAN [25], but they require a preset number of clusters. In actual scenarios, the number of staying and moving states of each user is uncertain, so it is difficult to preset a fixed value. Therefore, Tan *et al.* [23] proposed a clustering method based on spatiotemporal features to extract travel ODs, but when judging whether the trajectory point is staying or moving, it is still necessary to set fixed spatiotemporal thresholds. Kang *et al.* [26] proposed an algorithm to autonomously determine the number of clusters and can identify the significant places based on a time duration threshold (300 seconds). This method was less affected by the number of trajectory points, but it cannot be directly applied to identify the staying point of CSD. Liu *et al.* [27] proposed a mobility-based clustering method that used speed information to infer the degree of congestion of moving objects. However, this well-designed method is not suitable to capture the real staying clustering properties from the CSD used in our study. The main reason is that the real CSD are sparse which cause many clustering methods hard to extract the individual travel characteristics. And the second reason is that there is no unified definition of staying points, so it is not interpretable to define the staying points with fixed thresholds.

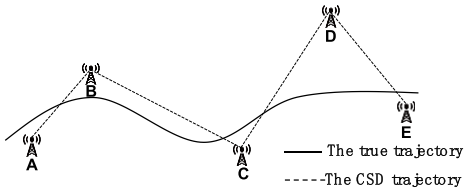

Fig. 2. Schematic diagram of the drift data (point D is a drift point, and the other locations are normal points).

C. Methods Based on Machine Learning

In recent years, some scholars have started to employ machine learning algorithms to identify staying points. Gong *et al.* [28] combined the support vector machine and C-DBSCAN algorithms to identify the staying points of the GPS trajectory, which may be suitable for identifying activity stops in continuous GPS trajectories with a higher frequency of data points. Unlike GPS data, the frequency of CSD is uncertain, and CSD is sparse in actual scenarios. Wang *et al.* [29] used the naive Bayesian algorithm to classify the ship's historical trajectories. Similarly, Zhao *et al.* [30] used the naive Bayesian algorithm to classify the individual trajectories of CSD. They constructed a naive Bayes classifier through the direction angle and the diameter of the minimal covering circle and then distinguished the staying or moving state. This kind of method does not involve the manual setting of the spatiotemporal threshold and the number or density of clusters. However, the classifier training requires a large number of actual samples (i.e., CSD) that are difficult to collect manually on a large scale.

In summary, these methods both require a fixed and preset threshold or the number of clusters. In addition, due to insufficient real data, the common approach to accuracy assessment is to perform a qualitative evaluation by mapping the results directly with GIS (geographic information system) and judging manually. This kind of assessment method is not convincing or reliable.

III. Methodology

Compared with other spatiotemporal data, CSD have considerable redundancy and error information, so it is necessary to carry out a comprehensive data cleaning operation before mining its application value. In this section, we first briefly introduce the process of data cleaning and then elaborate on the principle and implementation steps of the proposed staying point recognition algorithm.

A. Data Cleaning

The main idea for cleaning CSD is hierarchical cleaning, which considers the characteristics of the CSD (i.e., sparse data, ping-pong handover data, and drift data [31]). In this study, based on this idea, we conducted the following cleaning steps.

1) *Delete the records with missing values and the duplicate records*
2) *Merge the same location records*
3) *Process the drift records*

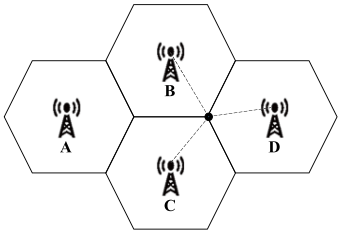

Fig. 3. Schematic diagram of the ping-pong handover data.

TABLE I
Interpretation of Parameters in Algorithm 1

Parameter	Interpretation
j	Sequence number of the spatiotemporal trajectory points, $j = 1,2, \ldots, n$. n is the total number of trajectory points
c	Number of clusters formed by clustering, $c = 1,2,3, \ldots$
$area$	Cluster area formed by spatiotemporal trajectory points
S_0	base spatiotemporal cost
m	Intermediate variable, each time a cluster is formed, the value of m changes

Calculate the speed between the adjacent records, and remove the records whose speeds are greater than 120 km/h (i.e., the speed limit).

1) Process the Ping-Pong Handover Records: As shown in Fig. 3, when a user is at the junction of base stations B, C, and D, the cell phone signal may switch back and forth among them since the signal strengths of the cell phone corresponding to the three base stations are similar. Considering that signal strength is related to the distance between cell phones and base stations, it is more likely that the user is located at the centroid of the base stations. In this situation, we calculate the average coordinate of consecutive ping-pong handover records and replace the original coordinates.

B. Staying Point Recognition

To avoid setting spatiotemporal thresholds and the number of clusters manually and empirically, we proposed a new algorithm considering the travel characteristics of individual one-day CSD to self-adaptively divide the CSD trajectory into staying clusters and moving clusters. It can be concluded from the CSD trajectory that the spatiotemporal characteristics of the data are significantly different when the traveler is in the state of staying or moving. Specifically, the data in the state of staying are more concentrated and have a higher density, while the data in the state of moving are the opposite. Based on the above characteristics, we hope to design a data container to clearly describe the two types of travel characteristics and build an iterative learning mechanism to obtain a reasonable number of containers.

As shown in Fig. 4, we describe the container as cuboids, and the core idea is to utilize flexible cuboids to encircle the same spatiotemporal points. The polyline in the figure is a trajectory from an individual daily CSD. Supposing a trajectory can be contained with n cuboids, whose size is freely flexible, the volume sum of each cuboid could be the maximum if

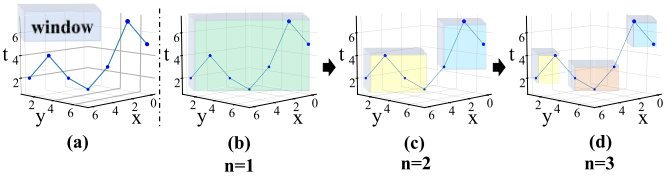

Fig. 4. Schematic diagram of the proposed algorithm.

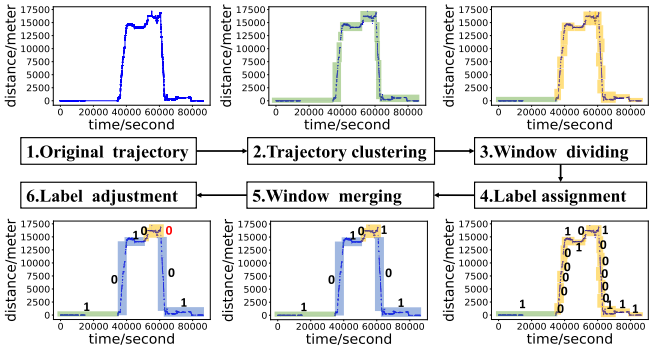

Fig. 5. Detailed process of the proposed algorithm from a 2D perspective.

$n = 1$; however, by adjusting the size of each cuboid, the sum exceeds the minimum, where $n = cubiod\ number$ and the process of clustering division stops. The volume of cuboids here can be understood as an entity with specialized spatiotemporal characteristics in which the points inside have the same attributes.

The idea of the algorithm consists of two parts. First, N cuboids are prepared to encircle an individual daily travel trajectory. Then, an iterative calculation starts from $N = 1$, and the iterative process continues by $N = N + 1$ if the volume of the cuboid decreases. An instance is shown in Fig. 4(b), (c), and (d), indicating that N changes from 1 to 3. In practical applications, we convert the 3D spatiotemporal trajectory into a 2D spatiotemporal trajectory to simplify the computational complexity. Fig. 5 shows the detailed process of the algorithm from a 2D perspective of a user's signaling trajectory, where "the rectangular box" is the spatiotemporal window (trajectory cluster), "1" is the staying cluster, "0" is the moving cluster, and the horizontal axis represents the time, the vertical axis represents the distance between each record and the first record after data cleaning.

1) Trajectory Clustering: Clustering is the basis of the algorithm, which divides an individual daily trajectory into several fundamental units with prominent spatiotemporal attributes. In this process, "base spatiotemporal cost" S_0 is introduced to describe the cost of every trajectory point. Thus, the spatiotemporal window area for each point is initially calculated according to the data characteristics. Then, the minimal area made from the consecutive points N is then calculated in chronological order and compared with $N * S_0$. Finally, several windows are confirmed in accordance with the comparison results, and thus, the fundamental clusters are obtained. The detailed process is described as follows, and the clustering process is shown in Algorithm 1.

Algorithm 1 Clustering Algorithm

```
Input  : PP: an individual's daily CSD trajectory; S0: "base spatiotemporal cost"
Output : PPc={ci}: Trajectory clusters
1  begin
2    Sort PP by time in ascending order;
3    m = 0;
4    PPc; = {}
5    j=0;
6    for j in Len (PP) do
7      Calculate area of the PPm:j+1
8       i= 0
9       if area> (j+2−m) ×S0:
10        pointm:j+1∈ci
11        Append ci to PPc;
12        m = j +1
13        i++
          —
14      j ++;
        —
15   return PPc
16 end
```

a) Considering the characteristics of individual daily trajectories, the base spatiotemporal cost S_0 is calculated by multiplying the personal average travel time and distance, which can be expressed as

$$S_0 = \frac{D_{mean} \times T_{mean}}{2} \tag{1}$$

where D_{mean} is the average travel distance among all the trajectory points and T_{mean} is the average staying time in every base station.

b) A serialization operation is performed to order the data chronologically. The result is expressed as $T_i = [t_1, t_2, \ldots, t_n]$, where n is the number of trajectory points.

c) Supposing there are m points from t_1 to t_m consisting of a spatiotemporal window (denoted as $cluster_0$), then the area can be obtained by using

$$area_0 = D_0 \times T_0 \tag{2}$$

where T_0 is the time difference between the first record and the last record in $cluster_0$ and can be calculated as $T_0 = time_m - time_1$, and D_0(intra-cluster distance) is the distance between the points formed by the maximum and minimum longitude and latitude of the base stations, which can ensure that all the points in the formed trajectory cluster are within the range of the spatiotemporal area and can be calculated as D_0, shown at the bottom of the page, where R is Earth's radius and $lat1$ and $lon1$ are the minimal latitude and longitude of the base

$$D_0 = 2R \times \sin^{-1}\left(\sqrt{sin^2\left(\frac{lat2 - lat1}{2}\right) + \cos(lat1) \times \cos(lat2) \times sin^2\left(\frac{lon2 - lon1}{2}\right)}\right),$$

station in $cluster_c$, respectively. $lat2$ and $lon2$ are the maximal latitude and longitude, respectively.

d) Rules are made to divide the trajectory into many clusters. It is based on the perception that if a newly added point has the same property as the old cluster, then its position in the spatiotemporal coordination system should appear to be close or inside the coverage of the window, which adds the spatiotemporal area as little as possible to make the characteristics of the cluster more prominent. In detail, if the spatiotemporal area of $cluster_0$ is smaller than $m * S_0$, the cluster continues to embrace the next point, and the process continues until the area of the new cluster becomes larger than $(m + 1) * S_0$.

e) Repeat steps c) and d) until all the trajectory points have been classified into suitable clusters.

2) Window Dividing: The clustering step is an efficient operation that divides the trajectory in a single traversal, but it may cause some clustering results not to satisfy the minimal area condition and thus influence the accuracy of trajectory point division. Therefore, the division operation should be carried out for the uniformity of each cluster. Since the average coverage radius of the 4G base station is approximately 300 meters [33] or 500 meters [32], thus the range of three base stations could contain a range of 1-kilometer individual traveling activity approximately. However, it is not easy to uniquely identify the types of activities in this situation, including staying, moving, and hovering states. To avoid the situation in which one cluster contains multiple types of activities, further division operation continues executing if the number of trajectory points in a cluster is more than three (see Algorithm 2).

Algorithm 2 Dividing Algorithm

```
Input  : S0: "base spatiotemporal cost"; PPc = {ci}: Trajectory clusters
Output : PPc~ = {c~i} : Trajectory clusters after dividing
begin
    i = 0
    PPc~ = {}
    j = 1
    for ci in PPc do
        m = 0
        for pointj in ci do
            Calculate area of the point0:j+1
            ΔSj = area − (j+1) × S0
            if ΔSj < ΔSj+1:
                m = j
                pointm:j+1 ∈ c~i
                Append c~i to PPc~;
                i++
            j = j + 1
        pointj+1:end ∈ c~i
        Append c~i to PPc~;
        i++
    return PPc~
end
```

Fig. 6 shows an example of the division algorithm. After clustering, six trajectory points in the trajectory cluster need

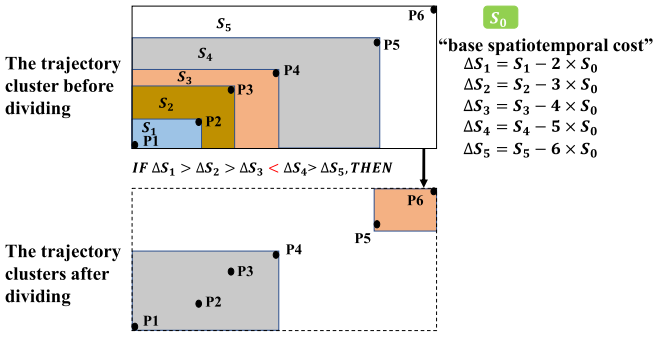

Fig. 6. Example of the dividing algorithm.

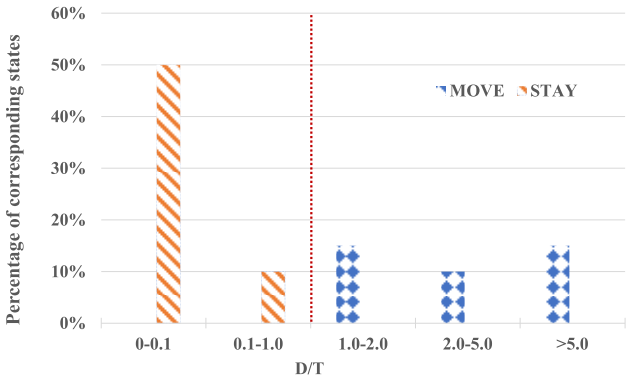

Fig. 7. Comparison of D/T in moving and staying states.

to be divided. We calculate the spatiotemporal area (S_1) formed by *P1* and *P2* and the spatiotemporal area (S_2) formed by trajectories *P1*, *P2*, and *P3*. Then we calculate the difference sequence between the spatiotemporal areas ($S_1, S_2, S_3, S_4, S_5, S_6$) and the base spatiotemporal cost S_0. If the difference sequence increases, the division stops; if not, the trajectory cluster is divided at the inflection point. After implementing the division algorithm, the original trajectory cluster is divided into two trajectory clusters, and the trajectory points in the divided trajectory cluster have higher cohesion.

3) Label Assignment: After obtaining the windows, we add labels to the windows based on the spatiotemporal characteristics of the surrounding trajectory points to distinguish the staying and moving states of each divided cluster. Intuitively, a spatiotemporal window is suggested to be marked as "moving" if the intra-cluster distance (D) of the window is greater than its stay interval (T). The window is marked as "staying" if the stay interval (T) is greater than the intra-cluster distance (D). The spatiotemporal properties of each cluster are defined as follows:

$$Cluster\ label = \begin{cases} 1, & D < T \\ 0, & D \geq T \end{cases},$$

where "1" denotes "staying" and "0" denotes "moving."

To verify our intuition, we calculate the D and T of the cluster formed by the trajectory points of known real staying and moving states and compare the D/T values in the two states, as shown in Fig. 7. The separator line is located

near 1, which means that it is feasible to label the clusters by comparing D and T.

4) Window Merging: Since the trajectory points of CSD include the staying state and moving state, the spatiotemporal windows with the same state should be merged to form a consecutive and intact individual trip chain. Therefore, we merge the windows with the same spatiotemporal label and then remark on the new combined window (see Algorithm 3).

Algorithm 3 Merging Algorithm

```
Input  : PP_c̃ = {c̃_i}: Trajectory clusters after splitting
Output : PP_c̃̃ = {c̄̄_j : Attribute_j}: Trajectory clusters after merging
begin
    i= 1, j=0, m=0
    PP_c̃̃ = {}
    for c̃_i in PP_c̃ do
        if Attribute_c̃i ≠ Attribute_c̃i-1 do
            cluster_c̃m:c̃i-1 ∈ c̄̄_j
            Calculate D_j and T_j of c̄̄_j
            if D_j > T_j
                Attribute_j = 0
            else: Attribute_j = 1
            Append c̄̄_j to PP_c̃̃;
            j++; m = i
        i++
    cluster_c̃m:c̃end ∈ c̄̄_j
    Append c̄̄_j to PP_c̃̃
    return PP_c̃̃
end
```

5) Label Adjustment: The initial label of each spatiotemporal window is determined by the rates of the window's length and width (i.e., the moving speed). In the case that a person walks 400 meters, his/her state should be labeled moving. However, because the coverage area of the base station in urban regions is approximately 500 meters [32], the cluster of the person's trajectory points is regarded as a candidate staying window to contain the scope of activity in the base station's coverage. Therefore, there may be some moving-state clusters misidentified as candidate staying windows.

To improve the accuracy of identifying the staying clusters, we adjust the labels by comparing the cluster's moving speed with the threshold of 0.5. If the stay interval of the candidate staying cluster is less than 5 minutes or the distance between the first base station in the cluster and the last base station in the previous cluster is greater than 400 meters [34], the candidate staying cluster is regarded as a moving cluster and relabeled as "moving".

6) Complexity of the Proposed Algorithm: This manuscript mainly involves three core algorithms. The complexity of Algorithm 1 is $O(nlogn)$, the complexity of Algorithm 2 is $O(n)$, and the complexity of Algorithm 3 is $O(m)$, where n is the number of individual daily CSD travel trajectory points, and m is the number of trajectory clusters formed by Algorithm 2. Therefore, the complexity of the proposed algorithm is $O(nlogn)$.

TABLE II
DATA COLLECTED BY APP

isdn	base_lng	base_lat	date	state
863252	113.387	23.0611	2020/01/04 0:00:02	stay
863252	113.309	23.0912	2020/01/04 10:00:22	move
863252	113.387	23.0611	2020/01/04 23:00:02	stay

TABLE III
ACTUAL NUMBER OF STAYS IN VALIDATION DATASET

Group	Actual number of stays	Note
001	4	Hovering
002	3	Hovering
003	5	Hovering
004	5	
005	5	Hovering
006	4	

IV. EXPERIMENT

A. Data Collection

To evaluate the accuracy of the proposed algorithm, we developed a smartphone application for collecting CSD and GPS data. This application can automatically record the information for the current base station communicating with the cell phone and simultaneously record the current GPS coordinates with a sampling frequency of 5 seconds. The GPS coordinate data, describing a real and complete individual trajectory, were used to assess the accuracy of the data cleaning results. Examples of the data collected by the application are shown in Table II.

Then, we recruited twelve volunteers to collect data for verification and divided them into six groups. As shown in Fig. 8, for the designed trip purposes (e.g., commuting, shopping, and entertainment), the volunteers used the application, traveled to different destinations along the planned routes with various trip modes (e.g., buses and subways), and stayed at the destinations for a period. Once the state changed, the volunteers manually recorded their state with the application. For example, when a volunteer planned to travel from the school to the park, the state had to be marked as "moving" before the trip and then marked as "staying" after arriving at the park. During some periods of staying, the volunteers were allowed to hover, i.e., they moved around in a particular area (e.g., parks and shopping malls). The number of staying states and moving states of each volunteer are shown in Table III.

B. Data Preprocessing

Taking the CSD of Group 005 as an example, we removed the incorrect and redundant data in the original CSD trajectories and retained all the essential spatiotemporal information for staying point recognition. From Fig. 9, the result is similar to the GPS trajectory, indicating that the method of data cleaning is effective. The original CSD trajectories have

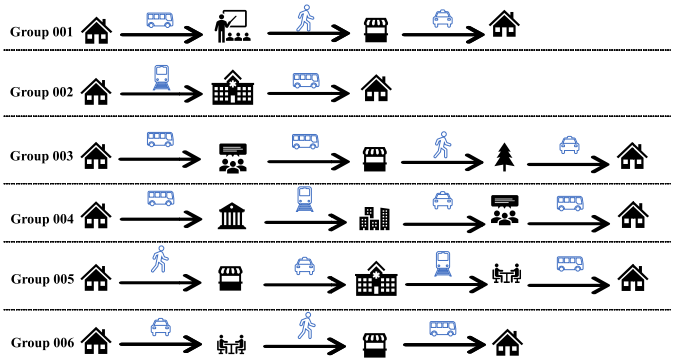

Fig. 8. Schematic diagram of the trajectories of volunteers.

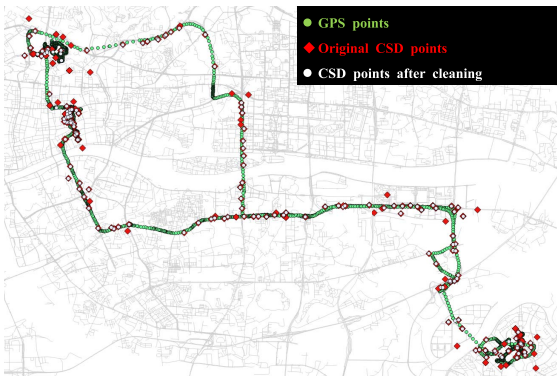

Fig. 9. Comparison chart before and after data cleaning.

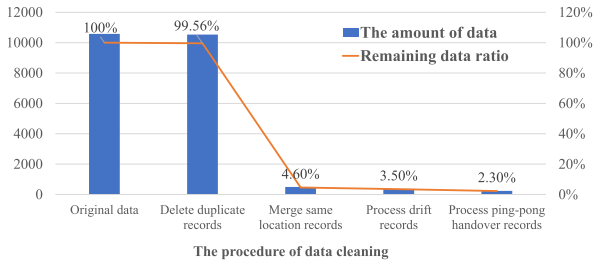

Fig. 10. Amount of data remaining after cleaning of CSD of group 005.

10,582 records, and the result after data cleaning has only 236 records, as shown in Fig. 10.

C. Results Analysis

To assess the accuracy of the proposed algorithm, we used four metrics, accuracy (A), precision (P), recall (R), and F1-score (F1), which are widely used in the accuracy assessment of classification methods.

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, F1 = \frac{2 \times R \times P}{P + R} \tag{4}$$

where TP and TN denote the times of correctly identified staying states and correctly identified moving state, respectively. FP denotes the times when a moving point is mislabeled as a staying point, while FN denotes the times when a staying point is mislabeled as a moving point. The results of each trajectory are shown in Table IV. The last column shows the total recognition accuracy of the six groups of trajectories and the calculation results of various indicators. The accuracy of the proposed algorithm is 91.3%. Fig. 11 shows the confusion matrix for the six groups of trajectories. The recall of 100% indicates that there is no staying state mislabeled as a moving state.

TABLE IV
RECOGNITION AND EVALUATION INDEX OF STAYING POINTS OF SIX GROUPS OF TRAJECTORIES

Group	Accuracy	Precision	Recall	F1-score
001	85.71%	80.00%	100.00%	88.89%
002	80.00%	75.00%	100.00%	85.72%
003	88.89%	83.33%	100.00%	90.91%
004	100.00%	100.00%	100.00%	100.00%
005	88.89%	83.33%	100.00%	90.91%
006	100.00%	100.00%	100.00%	100.00%
Total	91.30%	86.67%	100.00%	93.05%

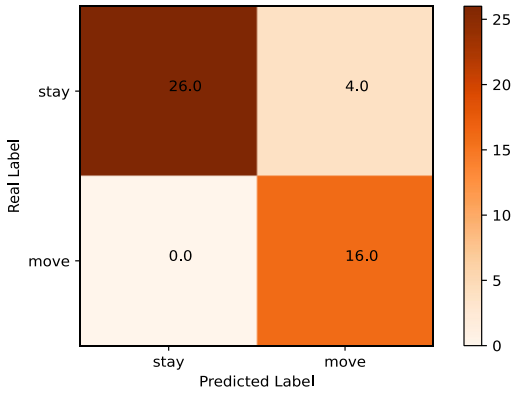

Fig. 11. Confusion matrix.

TABLE V
COMPARISON OF THE ACTUAL NUMBER OF STAYS AND THE NUMBER OF STAYS IDENTIFIED BY OUR ALGORITHM

Group	Actual number of stays	Number of stays in this algorithm	Number of matches
001	4	5	4
002	3	4	3
003	5	6	5
004	5	5	5
005	5	6	5
006	4	4	4

1) Number of Stays: From Table V, we find that the proposed algorithm tends to overpredict the number of stays. There are two main reasons. The first is the occurrence of moving states mislabeled as short-term stays (e.g., 5-minute

TABLE VI
ERROR RATE OF THE DURATION OF STAYS IDENTIFIED BY OUR ALGORITHM

Group	S1	S2	S3	S4	S5
001	16.25%	0.00%	10.77%	6.67%	NaN
002	1.13%	17.14%	4.76%	NaN	NaN
003	1.67%	23.75%	33.33%	35.00%	16.30%
004	3.33%	18.09%	21.25%	11.43%	7.27%
005	0.52%	24.40%	25.00%	3.08%	1.72%
006	1.42%	5.93%	0.75%	6.25%	NaN

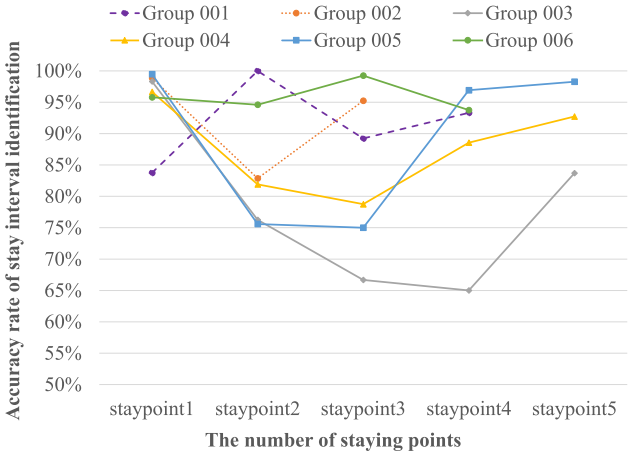

Fig. 12. Recognition accuracy of the stay interval of six groups of trajectories.

stays) due to the base station's broad coverage. The second is that hovering states were identified as multiple staying states.

2) Duration of Stays: To illustrate the recognition results for the duration of stays, we calculated the error rate of the duration of stays, which can be calculated as $error\ rate_{time}$, shown at the bottom of the page, where $endtime_a$ and $starttime_a$ are the end time and start time of the stay identified by the algorithm, $endtime_t$ and$starttime_t$ are the actual end time and start time of the stay, and $time_t$ is the actual duration of stay. The calculation results are shown in Table VI. S1–S5 represent the first to fifth stays, and 'NaN' indicates no stay.

Fig. 12 shows the accuracy of the recognition of duration of stays of the six groups of trajectories. The recognition accuracy of the duration of stays is greater than 60%, and the maximum is 100%. Furthermore, the accuracy of identifying the duration of stays in the morning and evening is higher (staypoint1 and staypoint5) because the users stayed in their place of residence during these periods.

3) Position of Stays: In addition to the number of stays and the duration of stays, the accuracy of the stay position is also one of the key indicators for evaluating the accuracy of staying point recognition. We mapped the staying points identified from each travel trajectory in Fig. 13 and compared them with the actual staying positions.

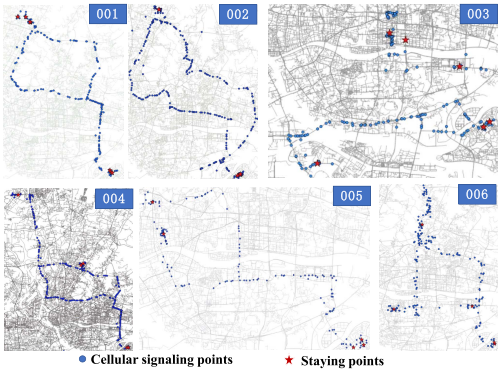

Fig. 13. Recognition results of the stay position of six groups of trajectories (the red stars represent the aggregated points).

TABLE VII
ACCURACY COMPARISON WITH SEVERAL METHODS

Method	Accuracy	Precision	Recall	F1-score
method1	81.48%	80.00%	85.71%	82.76%
method2	77.67%	70.67%	91.67%	79.77%
method3	70.83%	70.00%	85.19%	76.85%
proposed method	91.30%	86.67%	100.00%	93.05%

TABLE VIII
THE MEAN OF DURATION ERROR COMPARISON WITH SEVERAL METHODS

Method	The mean of duration error
method1	39.91 min
method2	35.45 min
method3	33.50 min
proposed method	20.92 min

D. Comparison of the Methods

We compared the proposed algorithm with the three types of staying point recognition method, including the method based on the fixed threshold [23], the method based on DBSCAN clustering [25], and the method based on naive Bayes classification [30], marked as method1, method2, and method3, respectively.

We used the CSD data from six groups of trajectories collected in the experiment and the metrics (accuracy, precision, recall, F1-score) to evaluate the recognition accuracy. From the comparison shown in Table VII, it can be seen that the proposed method has a better performance. We also compared the accuracy of the four methods in the duration of stays. From Table VIII, it can be seen that the proposed algorithm has the lowest mean duration error.

$$error\ rate_{time} = \frac{|endtime_a - endtime_t| + |starttime_a - starttime_t|}{time_t}$$

TABLE IX
COMPARISON RESULTS OF GROUPS IN DIFFERENT S_0

Group	S_0	Accuracy	Maximum accuracy
001	38210	85.71%	85.71%
002	54515	80.00%	100.00%
003	145410	88.89%	88.89%
004	198270	100.00%	100.00%
005	148210	88.89%	88.89%
006	201007	100.00%	100.00%

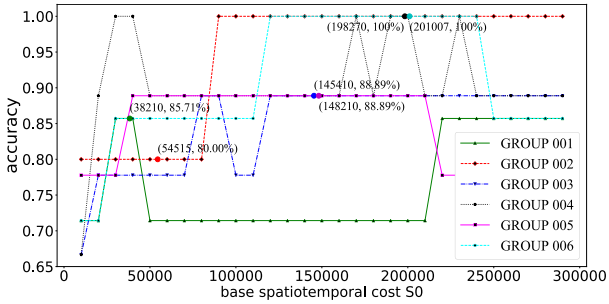

Fig. 14. Accuracy comparison of groups at different S_0.

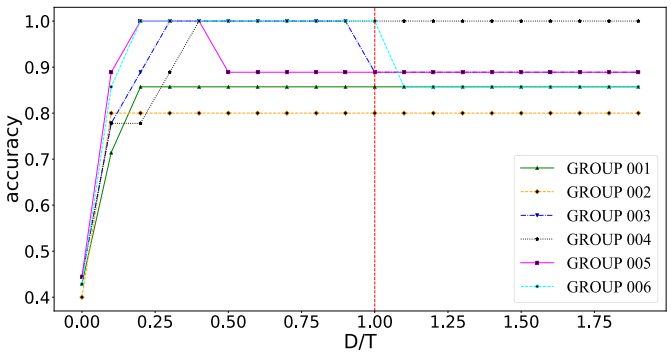

Fig. 15. Accuracy comparison of groups at different D/T.

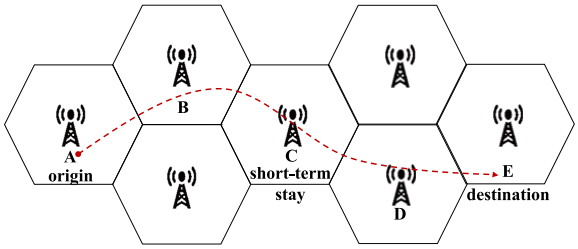

Fig. 16. Schematic diagram of short-term stay.

E. Sensitivity Analysis

In this section, we performed a sensitivity analysis on the proposed algorithm. The accuracy of the proposed algorithm is affected by the data quality related to factors such as the environment, signal strength, obstructions, communication corporations, and cell phone brands. These factors lead to missing data and the generation of sparse, drifting, and ping-pong handover data. To this end, a data cleaning algorithm was used to reduce its impact. According to the description of the algorithm in Section III-B, the factors affecting the performance of the algorithm include the base spatiotemporal cost S_0 and the value of D/T.

We set different S_0 and D/T values to compare the performance of the algorithm. Table IX shows the performance results of different groups of data. The value of S_0 is calculated using the average travel time and distance of the individual. The maximum accuracy represents the optimal performance of the algorithm under the group. Except for Group 002, the other groups achieve the best performance. From Fig. 14, the proposed algorithm is affected by the fluctuation of S_0, and it is difficult to determine a fixed S_0 for individuals due to the uniqueness of individual travel trajectories.

Fig. 15 shows the performance of the algorithm with different D/T values. Obviously, the proposed algorithm is affected by the D/T value. The D/T value in this study is set as 1 for two main reasons. First, in the actual situation, the D/T value describes the speed of the user, and the value of 1 m/s (i.e., $D/T = 1$) is a typical value of walking speed. Second, it is difficult to set a value of D/T for each user in practice.

V. DISCUSSION

A. Hovering and Short-Term Stays

The proposed algorithm can accurately identify the staying point in most cases. In the case of recognizing hovering stays, this kind of stay was identified as multiple stays. To facilitate the extraction of individual trip chains, we aggregated these stay points into a single stay point whose time was the sum time and the location was the mean of the points.

However, for short-term stays, they cannot be effectively identified due to the generation mechanism of CSD. For example, as shown in Fig. 16, there are 5 base stations (A, B, C, D, and E). A person traveled from A to E and stayed at C for a short period. Theoretically, the recorded CSD data should include the locations of the five base stations. Due to the instability of communication, there may be less than 1 record generated from base station C, which is insufficient for identifying this kind of stay.

B. Sparsity of CSD Data

In actual scenarios, the individual CSD are usually very sparse and inaccurate due to the uncertainty in the communication between cell phones and base stations. The existing method can deal with the problem caused by the sparsity of GPS data [35], but it is not suitable for addressing the issue in terms of CSD. We conducted a statistical analysis of the CSD data from some users in Foshan, China. The data sampling interval ranges from 0 to 400 seconds, and the average sampling interval is 370 seconds, as shown in Fig. 17. The average number of CSD points per user per day is 194, as shown in Fig. 18.

To ensure that the collected CSD data are sufficient and reliable, the application's sampling frequency is 5 seconds. The number of data points for each user each day is 7,240 on average. To assess the performance of the proposed algorithm in actual scenarios, we adopted a resampling method. We randomly selected the data at different rates from 5% to 100% and used the proposed algorithm to identify the staying points from the selected data. As displayed in Fig. 19, the proposed

TABLE X
COMPARISON RESULTS OF CSD AND GPS STAYING POINTS RECOGNITION

Group005	Accuracy	Precision	Recall	F1-score
CSD	88.89%	83.33%	100.00%	90.91%
GPS	81.82%	75.00%	100.00%	85.71%

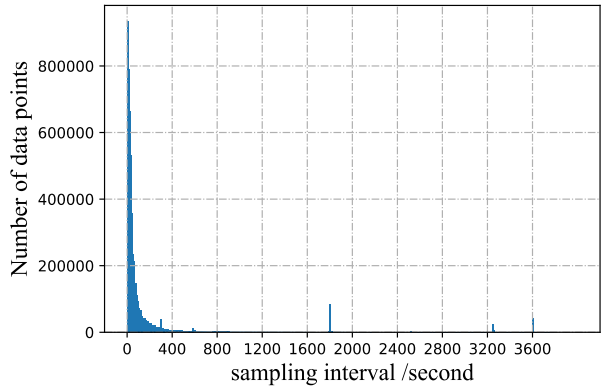

Fig. 17. Histogram of the real CSD data sampling interval distribution.

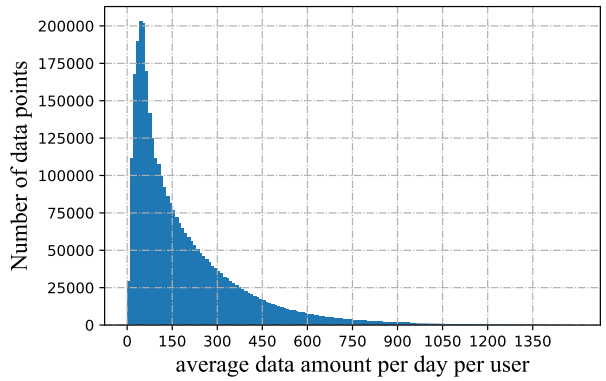

Fig. 18. Histogram of average data amount per day per user distribution.

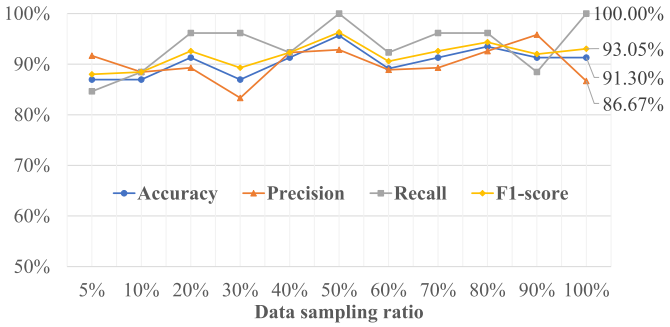

Fig. 19. Performance comparison of four indicators at different sampling rates.

algorithm still performs well for 5% of the data, which is close to the general data size.

In addition, the proposed algorithm also has good performance on another dataset, i.e., GPS data. The application collects CSD data and GPS data simultaneously. A sample of these data is shown in Fig. 10. We take the GPS of Group 005 as an example to calculate the staying points, as shown in Table X. The accuracy of GPS is lower than that of CSD, but the recall is also 100% because the GPS data records more spatial location points to identify short-term stays caused by traffic jams during movement. However, the volunteers did not record the stays caused by traffic jams during the experiment, which caused the GPS calculation to perform slightly worse than that of CSD.

VI. CONCLUSION

In this study, to avoid presetting spatiotemporal thresholds empirically as in previous work, we present a new algorithm of staying point recognition for identifying the staying points and moving points in cellular signaling data. The proposed algorithm, a kind of iterative-learning-based algorithm, can cluster trajectory points and label the clusters concerning the CSD characteristics, which offers a new perspective for spatiotemporal trajectory clustering.

Based on the ground truth of cellular signaling data collected by our developed application, we conducted empirical experiments, sensitivity analysis, and performance comparison to confirm the merits of the proposed algorithm. The results showed that our algorithm performed optimally and achieved an accuracy of 91.3%. From the sensitivity analysis, we found that the performance of the proposed algorithm was related to the base spatiotemporal area and the value of D/T (i.e., the speed of the user). Considering the individual characteristics, we adaptively set different base spatiotemporal areas for each user. However, it is difficult to set the value of D/T for each user, so we selected a fixed value considering the walking speed of a human.

According to our experimental analysis, it can be found that the proposed method can be applied not only to CSD but also to GPS data. Whether it is high-frequency trajectory data or sparse location data, this method has good adaptability. It can accurately identify the staying points of spatiotemporal trajectories, which provides a basis for the application of location data in transportation.

In future work, we will apply the proposed algorithm to a large amount of real cellular signaling data and study more application scenarios, e.g., additional trajectory data and the processing of short-term staying points with other data.

ACKNOWLEDGMENT

The authors would like to extend their heartfelt gratitude to Lan, Ziqin, and Yang, Yingkun for their proofreading and assistance on this paper.

REFERENCES

[1] F. Calabrese, L. Ferrari, and V. D. Blondel, "Urban sensing using mobile phone network data: A survey of research," *ACM Comput. Surveys*, vol. 47, no. 2, pp. 1–20, Jan. 2015.

[2] Y. Jin, "Reducing cellular signaling traffic for heartbeat messages via energy-efficient D2D forwarding," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, Jun. 2017, pp. 1301–1311.

[3] Z. Li, L. Yu, Y. Gao, Y. Wu, G. Song, and D. Gong, "Identifying temporal and spatial characteristics of residents' trips from cellular signaling data: Case study of Beijing," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2672, no. 42, pp. 81–90, Dec. 2018.

[4] E. Thuillier, L. Moalic, S. Lamrous, and A. Caminada, "Clustering weekly patterns of human mobility through mobile phone data," *IEEE Trans. Mobile Comput.*, vol. 17, no. 4, pp. 817–830, Apr. 2018, doi: 10.1109/TMC.2017.2742953.

[5] F. Wang and C. Chen, "On data processing required to derive mobility patterns from passively-generated mobile phone data," *Transp. Res. C, Emerg. Technol.*, vol. 87, pp. 58–74, Feb. 2018.
[6] D. Bachir, "Inferring dynamic origin-destination flows by transport mode using mobile phone data," *Transp. Res. C, Emerg. Technol.*, vol. 101, pp. 254–275, Apr. 2019.
[7] S. Colak, "Analyzing cell phone location data for urban travel: Current methods, limitations, and opportunities," *Transp. Res. Rec., J. Transp. Res. Board* vol. 2526, no. 1, pp. 126–135, 2015.
[8] X. Chen, "Trip-chain-based travel-mode-shares-driven framework using cellular signaling data and web-based mapping service data," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2673, pp. 51–64, Mar. 2019.
[9] K. Chin, "Inferring fine-grained transport modes from mobile phone cellular signaling data," *Comput., Environ. Urban Syst.*, vol. 77, Sep. 2019, Art. no. 101348.
[10] S. Jiang, J. Ferreira, and M. C. Gonzalez, "Activity-based human mobility patterns inferred from mobile phone data: A case study of Singapore," *IEEE Trans. Big Data*, vol. 3, no. 2, pp. 208–219, Jun. 2017.
[11] C. Qian, "Measuring spatial distribution of tourist flows based on cellular signalling data: A case study of Shangha," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 2584–2590.
[12] L. Ni, X. Wang, and X. Chen, "A spatial econometric model for travel flow analysis and real-world applications with massive mobile phone data," *Transp. Res. C, Emerg. Technol.*, vol. 86, pp. 510–526, Jan. 2018.
[13] G. Zhong, T. Yin, J. Zhang, S. He, and B. Ran, "Characteristics analysis for travel behavior of transportation hub passengers using mobile phone data," *Transportation*, vol. 46, no. 5, pp. 1713–1736, 2018.
[14] X. M. Chen, S. Zhang, and M. Zahiri, "Cellular signaling data driven simulation-based dynamic traffic assignment and its applications to a real-world road network," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 2083–2088.
[15] D. Bachir, V. Gauthier, M. E. Yacoubi, and G. Khodabandelou, "Using mobile phone data analysis for the estimation of daily urban dynamics," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 626–632.
[16] T. Tettamanti, H. Demeter, and I. Varga, "Route choice estimation based on cellular signaling data," *Acta Polytechnica Hungarica*, vol. 9, no. 4, pp. 207–220, 2012.
[17] M. Yin, M. Sheehan, S. Feygin, J.-F. Paiement, and A. Pozdnoukhov, "A generative model of urban activities from cellular data," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 1682–1696, Jun. 2018.
[18] C. Chen, L. Bian, and J. Ma, "From traces to trajectories: How well can we guess activity locations from mobile phone traces?" *Transp. Res. C, Emerg. Technol.*, vol. 46, pp. 326–337, Sep. 2014.
[19] X. Zhou, A. G. Yeh, W. Li, and Y. Yue, "A commuting spectrum analysis of the jobs–housing balance and self-containment of employment with mobile phone location big data," *Environ. Planning B, Urban Anal. City Sci.*, vol. 45, no. 3, pp. 434–451, May 2018.
[20] L. Ying, H. Baorui, and L. Yimin, "Bus scheduling feasibility study of rainy day based on the mobile phone signal data," in *Proc. Int. Conf. Intell. Transp., Big Data Smart City*, Hạ Long Bay, Vietnam, Dec. 2015, pp. 192–195, doi: 10.1109/ICITBS.2015.54.
[21] C. Horn and R. Kern, "Deriving public transportation timetables with large-scale cell phone data," *Procedia Comput. Sci.*, vol. 52, no. 1, pp. 67–74, 2015.
[22] J. L. Toole, S. Colak, B. Sturt, L. P. Alexander, A. Evsukoff, and M. C. González, "The path most traveled: Travel demand estimation using big data resources," *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 162–177, Sep. 2015.
[23] J. Tan, L. Dong, J. Gao, W. Guo, and Z. Li, "The methods of extracting spatiotemporal characteristics of travel based on mobile phone data," in *Proc. IEEE 7th Data Driven Control Learn. Syst. Conf. (DDCLS)*, May 2018, pp. 1174–1179.
[24] P. Widhalm, Y. Yang, M. Ulm, S. Athavale, and M. C. González, "Discovering urban activity patterns in cell phone data," *Transportation*, vol. 42, no. 4, pp. 597–623, Jul. 2015.
[25] S. Qin, Y. Zuo, Y. Wang, X. Sun, and H. Dong, "Travel trajectories analysis based on call detail record data," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, Chongqing, China, May 2017, pp. 7051–7056, doi: 10.1109/CCDC.2017.7978454.
[26] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello, "Extracting places from traces of locations," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 9, no. 3, pp. 58–68, 2005.
[27] S. Liu, Y. Liu, L. M. Ni, J. Fan, and M. Li, "Towards mobility-based clustering," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: ACM, 2010, pp. 919–928, doi: 10.1145/1835804.1835920.
[28] L. Gong, H. Sato, T. Yamamoto, T. Miwa, and T. Morikawa, "Identification of activity stop locations in GPS trajectories by density-based clustering method combined with support vector machines," *J. Mod. Transp.*, vol. 23, no. 3, pp. 202–213, 2015.
[29] W. Wang, X. Chu, Z. Jiang, and L. Liu, "Classification of ship trajectories by using naive Bayesian algorithm," in *Proc. 5th Int. Conf. Transp. Inf. Saf. (ICTIS)*, Liverpool, U.K., Jul. 2019, pp. 466–470, doi: 10.1109/ICTIS.2019.8883562.
[30] G. Zhao, J. Lai, Y. Chen, H. Sun, and Y. Zhang, "Residents' travel origin and destination estimation method based on the naive Bayes classification," *J. Comput. Appl.*, vol. 40, no. 1, pp. 36–42, 2020.
[31] V. Etter, M. Kafsi, and E. Kazemi, "Been there, done that: What your mobility traces reveal about your behavior," in *Proc. Mobile Data Challenge Nokia Workshop*, 2012, pp. 1–6.
[32] C. Horn, S. Klampfl, M. Cik, and T. Reiter, "Detecting outliers in cell phone data: Correcting trajectories to improve traffic modeling," *Transp. Res. Rec.*, vol. 2405, no. 1, pp. 49–56, Jan. 2014.
[33] F. Calabrese, G. D. Lorenzo, L. Liu, and C. Ratti, "Estimating origin-destination flows using mobile phone location data," *IEEE Pervas. Comput.*, vol. 10, no. 4, pp. 36–44, Apr. 2011.
[34] Highway Capacity Manual, "Transportation research board," Nat. Res. Council, Washington, DC, USA, Tech. Rep., 2000, ch. 4.1.4.
[35] L. Lu, N. Cao, S. Liu, L. Ni, X. Yuan, and H. Qu, "Visual analysis of uncertainty in trajectories," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2014, pp. 509–520.

Ming Cai received the B.S. and Ph.D. degrees from Sun Yat-sen University, Guangzhou, China, in 1999 and 2004, respectively. He is currently a full-time Professor with the School of Intelligent Systems Engineering, Sun Yat-sen University. His main research interests include the traffic big data, traffic environmental engineering, and intelligent transportation systems.

Zixuan Zhang received the B.E. degree from Jinan University, Guangzhou, China, in 2019. She is currently pursuing the M.S. degree in transportation engineering with Sun Yat-sen University, Guangzhou. Her research interest includes traffic big data.

Chen Xiong received the B.E. degree in industrial engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2012, and the Ph.D. degree in optical measurement mechanics from the University of Science and Technology of China, Hefei, China, in 2017. He carried out research in intelligent perception at Huawei Technologies Company Ltd., Shanghai, China. He is currently a Post-Doctoral Researcher with the School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou, China. His research interests include traffic big data and intelligent transportation systems. His work was supported by NSFC and MSTC.

Chao Gou received the B.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2012, and the Ph.D. degree from the University of Chinese Academy of Sciences (UCAS), Beijing, China, in 2017. From September 2015 to January 2017, he was supported by UCAS as a Joint-Supervision Ph.D. Student with Rensselaer Polytechnic Institute, Troy, NY, USA. He is currently an Assistant Professor with the School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou, China. His research interests include computer vision and machine learning.