

Research Article

Detecting Suspects by Large-Scale Trajectory Patterns in the City

Cang-Hong Jin , Dong-Kai Chen , Fan-Wei Zhu, and Ming-Hui Wu 

School of Computer & Computing Science, Zhejiang University City College, Hangzhou 310015, China

Correspondence should be addressed to Ming-Hui Wu; mhwu@zucc.edu.cn

Received 10 November 2018; Revised 24 April 2019; Accepted 24 June 2019; Published 10 July 2019

Academic Editor: Paolo Bellavista

Copyright © 2019 Cang-Hong Jin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A massive amount of spatial-temporal records generated by sensors across the city help describe our day-to-day activities. Since the lifestyle represented by moving data varies from one individual to another, data analysts could facilitate the suspect-detection task by analyzing and classifying related trajectories of a given target. However, there are still some challenges that need to be overcome in real-life cases; for instance, the positive instances are limited, the trajectories are too diverse, and the transit behavior features are both too broader and costly to define. Moreover, people living in different areas of the city may have different life habits which can result in incorrect conclusions due to data-sensitive factors. In this paper, we describe the particular characteristics of movement behaviors regarding trajectory features. We also propose two models to improve the identification performance, namely, the trajectory pattern model (TPM) and neural network-based model. The trajectory pattern model (TPM) offers a novel view to describe users' movement behaviors and generates more effective and universal features other than location and timestamp dimensions. The end-to-end neural network-based model aims to avoid picking human features. Statistical analysis and insightful explanations are provided to help understand the behavior of a given target. The effectiveness of our proposed solutions compared to peer solutions is demonstrated and proved via extensive evaluation.

1. Introduction

Crime suspect is a person who is suspected of committing a criminal offense and would potentially threaten the public safety of the community. Identification of suspects is an important yet challenging task for Public Security Department. With rapid development in sensor technology and data-processing capabilities, analyzing moving records becomes extremely helpful for understanding residents' mobility patterns. For instance, the police nowadays are more likely to turn to find crime suspects by seeking data from mobile phones in target areas on the Google Map [1]. In Beijing, researchers leveraged massive automated fare collection (AFC) transit records to identify pickpockets from regular passengers [2]. Other studies focus on classifying people by trip characteristics, such as the preferred travel choice and travel mode [3]. Understanding these movement patterns not only enables us to depict the lifestyles of various groups of residents but also gives us the insight to perceive the hidden patterns behind these trajectories. Although there are several spatial-temporal-based methods to detect

anomaly behaviors, it is still difficult to solve our problem because most of the suspects do not have distinguishing characteristics before they are arrested, and the raw sensor data are complicated. For example, (1) there are a great amount of none-mobile-phone terminal data that cannot present an individual's activities; (2) unlike data captured from automated fare collection systems which have a fixed routine, people traveling in the city may move randomly or irregularly which means the pattern of their motion is totally unpredictable; (3) the lifestyle of residents living in different areas of the city is not quite similar considering the variance of local transportation system preferences. Therefore, classifiers based on a special location or functional region cannot transfer to other scenarios; and (4) the number of suspects (positive instances) is very scarce and sparse, and to the best of our knowledge, there is no existing report describing suspects' movement patterns.

A solution to address the aforementioned issues is to identify suspects regardless of time and location. In this article, we present three models to extract transit features and use machine learning methods to verify the effectiveness

of the real-life dataset. We also developed an end-to-end learning model named “MST-CNN” to fully automate the identification process. In addition, to understand the transit behaviors of suspects, we analyze the characteristics of suspects and regular residents via visual-aided and statistical methods. The major contributions of this work are as follows:

- (i) We define the identification problem through massive transit records in the city and try to explain why the task is not trivial in the real-life scenario.
- (ii) We design the trajectory pattern model to present individuals’ behaviors and extract some effective features for classification and anomaly detection methods.
- (iii) Using the cleaned data, we introduce an end-to-end deep learning model to capture the task directly and demonstrate that our model outperforms peer methods in terms of precision and recall.
- (iv) We discuss the pros and cons of feature-based methods and a deep learning method, respectively. We also visualize the trajectories of two groups and explore the interesting phenomena of them.

2. Mobility Characteristics and Problem Definition

2.1. Moving Data and Related Information. The data in our paper are gathered from Wi-Fi sensors including moving points and sensor-related information. By utilizing moving data, we could draw the people’s trajectories across the city.

2.1.1. Moving Point. Wi-Fi sensors request signals that are continuously broadcast from Wi-Fi-enabled devices, such as smartphones, laptops, and tablets. When a client wants to connect to a wireless network, they will send probe requests including unique MAC address, brand, manufacturer, and model, which tell when the user passes within the range of our access points, for as long as the user is in the area.

2.1.2. Geographic Information. Each Wi-Fi sensor has a unique location and a special scenario. For example, the data collected by a sensor installed in front of a gate of a railway station could infer who is leaving or arriving at the city with high probability. Now the source data could be extended with richer information like $\langle \text{MAC}, \text{SSID}, \text{device address}, \text{longitude}, \text{latitude}, \text{scenario type} \rangle$.

2.2. Mobility Characteristics. We provide insights into the diversity phenomenon of residents’ movement patterns from three aspects as follows:

- (i) *Location Preference.* A city could be divided into different regions according to their functions as CBD and station. Here, we plot the heat maps of two

groups of people by their most frequently visited places in Figure 1 and find that the hot places are similar after removing people’s fixed residence. Thus, a special location might not be a good feature in our case, and our experiments also prove it.

- (ii) *Activity Scope.* We refer to statistics of users’ activity scope that vary with fuzzy region size, which is on the assumption that some people need to travel across the city to work and others prefer to work close to their home. We use the hash (len=5) method to fuzz the geographic information (called GeoLoc) and then calculate the amount of visited GeoLoc for everyone. In Figure 2, the travel scope represents the range of motion and the rate represents the popularity of each travel scope. It shows that residents mostly travel farther than suspects. One of the reasons is that many suspects who have low education and are unemployed or self-employed prefer to transit nearby.
- (iii) *Life Habits.* People have their habits of commuting and lifestyle in the city, which means people tend to do a routine of behavior regularly and sub-consciously. In order to estimate people’s life habits, we collect their GeoLoc in weekdays, as shown in Table 1. We define dayLoc as the set of regions visited in weekdays and weekLoc as the union set of dayLoc, and then the similarity could be calculated by the Jaccard index:

$$J(\text{dayLoc}, \text{weekLoc}) = \frac{|\text{set}(\text{dayLoc})|}{|\text{set}(\text{weekLoc})|} \quad (1)$$

The average similarity value sim of the user is $(0.6 + 0.4 + 0.4 + 0.5 + 0.5)/5 = 0.48$. A higher sim value indicates the user’s travel behavior is more predictable and less volatile. We use this indicator to measure the regularity of individuals’ behavior, and we can tell from the rate that how many people are regular in their activities when it comes to suspects or residents. Figure 3 illustrates that two groups have no significant difference in travel habits and either residents or suspects have a few regular patterns.

2.3. Problem Definition

Definition 1 (moving point). A location with a corresponding timestamp and MAC is named a moving point $O = (p, m, t)$, where p is a location with latitude and longitude, m is a unique MAC address, and t is a timestamp.

Definition 2 (path). Given a set of moving points O and a special MAC a , a path of certain people could be defined as $P = \{O_1, O_2, \dots, O_n\}$, where $\forall O_i(m) = a$ and $O_i(t) < O_j(t)$ when $i < j$.

A path may include various trips and span a long time. And still, MAC would be collected by the same sensor repeatedly to generate noise data, which will make trouble in our model. Therefore, we need to split the path into

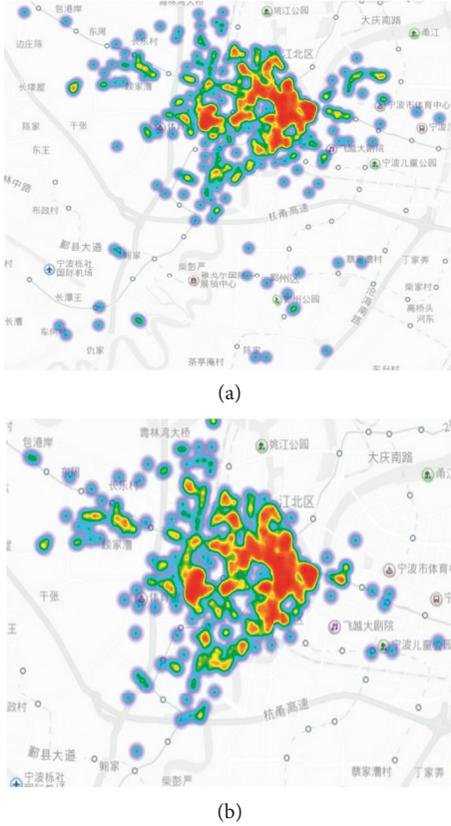


FIGURE 1: Activity places of (a) suspects and (b) residents.

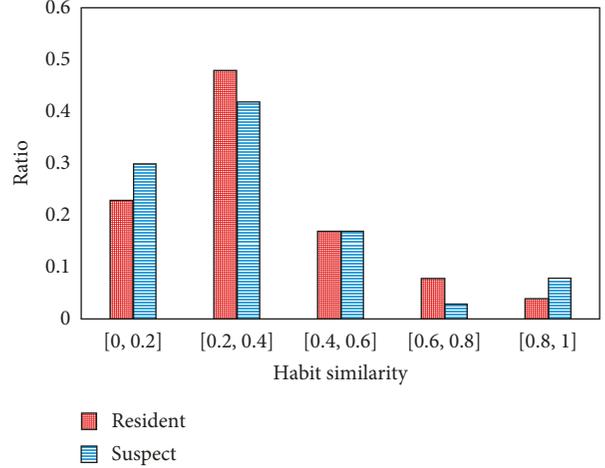


FIGURE 3: Travel habits of different users.

trajectories and maintain the first and the last record for each sensor in continuous time.

Definition 3 (trajectory). A trajectory T is a subset of a path with a clear scenario. We define two criterions to generate a trajectory:

- (1) $O_i(t) - O_{i-1}(t) < \tau$
- (2) $\text{distance}(O_i(p) - O_{i-1}(p)) > d$, where i is the interval time and d is the minimal distance

Problem 1 (suspect identification). Given a user u and his/her relevant trajectories T_u^i during a period of time, it is detected whether u is a suspect or not.

2.4. Trajectory Compression. Transit data are becoming increasingly available, and the size of recorded trajectories is getting larger. Thus, we need to compress planar trajectories such that the most common spatial-temporal semantic can still be maintained approximately after the compression has taken place. We compress our moving data by the Douglas-Peucker algorithm, which uses a point-to-edge distance tolerance. It can compute the distance of all intermediate vertices to the edge, which would be used to compress the trajectory. In order to match the error of sensors, we apply the technique of sliding window in Algorithm 1, and this can be seen in Figure 4.

In practice, we construct a trajectory record if and only if the time gap between two moving points is no more than 30 minutes. Then, the trajectory is compressed with a ϵ value of 200 m.

3. Feature-Based Identification Framework

In this section, we introduce the key components of our identification framework in Figure 5. Especially, in order to distinguish suspects from regular residents, we first propose three feature selection models in multiple levels: *static*

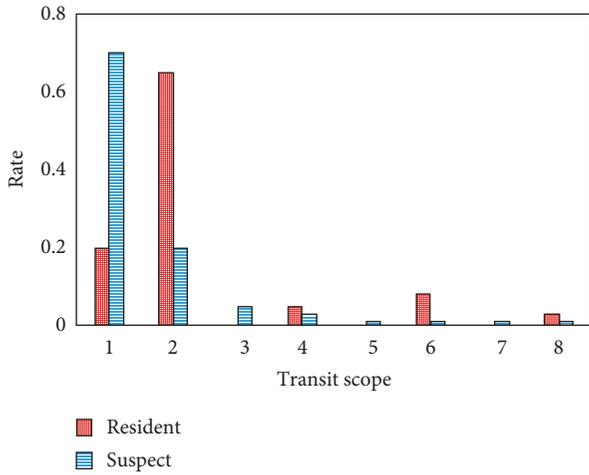
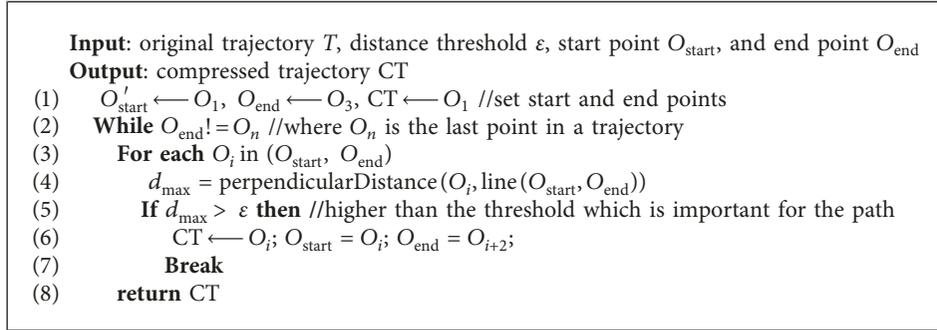


FIGURE 2: Travel scope of different users.

TABLE 1: Similarities of users' movement habits.

User	Day	GeoLoc sequence	Similarity
m_1	Monday	$P_1, P_2, P_3, P_4, P_6, P_7$	0.6
m_1	Tuesday	P_2, P_3, P_5, P_8	0.4
m_1	Wednesday	P_2, P_4, P_9, P_{10}	0.4
m_1	Thursday	P_5, P_6, P_7, P_8, P_9	0.5
m_1	Friday	P_1, P_3, P_4, P_7, P_9	0.5



ALGORITHM 1: Transit compression algorithm (TCA).

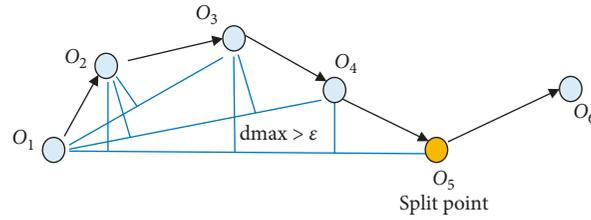


FIGURE 4: Trajectory compression process.

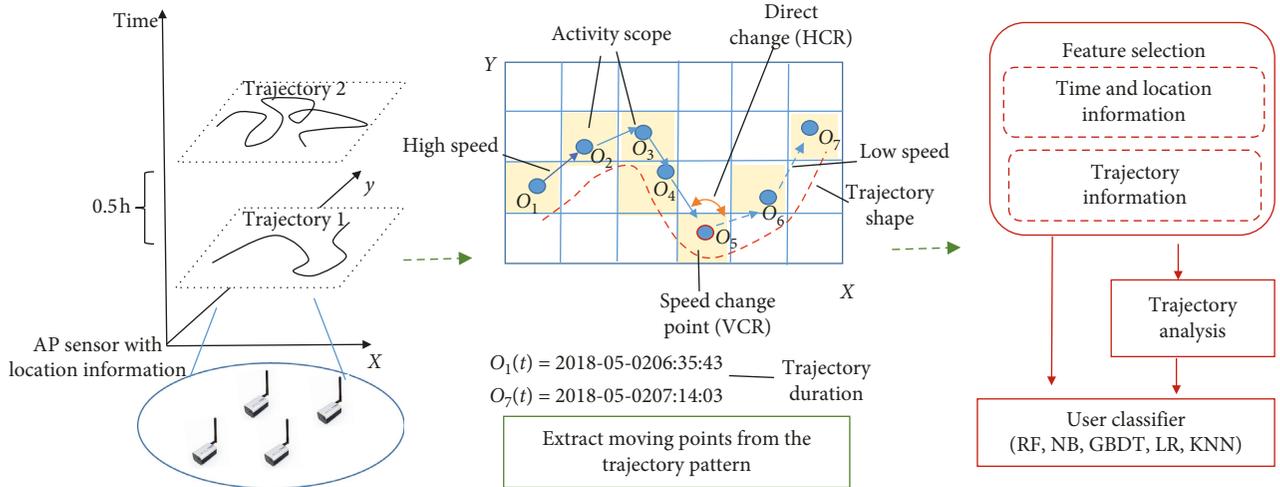


FIGURE 5: Mining process and feature description of the trajectory model.

regions, track change points, and trajectory features. Then, we use classification and anomaly detection methods to evaluate the performance with extracted features.

3.1. Static POI Model. The feature selection model is designed based on an assumption that places residents visited in the city might represent trip characteristics. A sensor contains both geographic and scenario information; thus, we can transfer a series of moving points to people's static POI model (SPM). Since our dataset has more than thirty thousand access points (APs) and the timestamp is accurate to milliseconds, we need to discretize the timestamp factor and merge geography factors to larger regions. For

time factors, we use discretized filters to obtain categorical attributes for every 15 minutes. Although we could divide numerical time into various discrete bins by different ranges (e.g., 5, 10, and 30 minutes), in our experiments, a 15-minute filter is better than other filters.

Then, we encode each AP sensor geographic information by Geohash, a hierarchical spatial data structure that encodes a geographic location into a short character string. Given latitude and longitude coordinates, Geohash produces a slightly short hash. Different lengths of Geohash can represent geographic information of different precision. In our study, we use Geohash of eight characters, whose accuracy error is 60 m. For instance, an object $O = (p, m, t)$ is transferred to

$O_{SPM} = (p_h, m, m_t, t_d, t_w)$, where p_h is the Geohash code of p , m_t is the MAC type (0 is normal and 1 is suspect), t_d is the day of timestamp, and t_w is the time-slice window of timestamps. For example, given a sensor object $\langle 1430042E10BC, 121.55494, 29.87406, 1525104275 \rangle$, we can encode it to $\langle wtq3yn, 1430042E10BC, 1, 2018-05-01, 1 \rangle$. Eventually, we get 96 time-slice window features and 117 encoded geofeatures in our SPM approach.

3.2. Location-Changed Model. Because of the inherent inaccuracies of the measurement devices (e.g., Wi-Fi sensor), it is not expected to find exact activity patterns by the SPM. For example, when a device is put at home but in the scan scope of the AP sensor, related records will be collected repeatedly all the time. This means that even if we detect position information, we cannot judge whether the person is still or moving. Even worse, a MAC could be detected by multiple sensors at the same time, which looks like a person travels among these sensors.

In order to extract dynamic records from the SPM, we promote the SPM by additional location transfer features, which would guarantee the object was exactly moving during the time. Three criterions are generated to define "location transfer," and Algorithm 2 is given as follows.

The input and output of the LCM are the same as those of the SPM, but we give up those stable records and extract moving points as useful data treated as classifier features. The motivation of the LCM is that we cannot exactly infer what time a user passes a point. If he/she is moving, then these continuous points are in sequential time, but if a person stays at a point for a long time, we cannot precisely infer which time slices he/she starts moving, and thus, we tend to believe that the last time slice of the current moving point is the time of the previous point, that is, what Line12 in the LCM is doing.

3.3. Trajectory Pattern Model. The above static point and active point features keep a view on geographic information and related timestamp. But a single moving point would miss some meaningful context of user behaviors. For example, the purpose of going to work in the morning or strolling in the gardens midnight would be reflected by separated points. Therefore, trajectories could reflect the movement behavior of the tracked object or phenomenon, from which we could extract behavior regular patterns. First, we extract personal trajectories from global moving points, and then for each track, we propose some movement features to describe the trajectory pattern.

Based on the above description, we give more explanation about the following 8 movement pattern features defined by us for each trajectory including activity scope, trajectory duration, trajectory length, speed, velocity change rate, head change rate, track curve, and activity pattern.

Activity scope measures the resident's movement area by the number of Geohash regions. Trajectory duration is the total transit time of the trajectory. Trajectory length is the length of the whole path, and curve can present the complexity of the path. We define the curve by the following formula:

$$\text{curve} = \frac{\sum_{i=2}^n \text{distance}(O_i, O_{i-1})}{\text{distance}(O_n, O_1)}, \quad (2)$$

where

$$\text{distance}(O_i, O_j) = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}. \quad (3)$$

In Figure 6, there are six points in a path with a starting point O_1 and an ending point O_6 , and the curve is the sum of the distance between adjacent points divided by the distance between O_1 and O_6 .

We use speed and velocity change rate (VCR) to infer the transit tool. The speed of two points $\text{Speed}(O_2, O_1)$ could be calculated by $\text{distance}(O_2, O_1)/\text{time}(O_2, O_1)$. Also, the head change rate (HCR) θ is to measure the direction changes of a single trajectory, as shown in Figure 7. The formula is

$$\alpha = \arccos \frac{a^2 + b^2 - c^2}{2ab},$$

$$\theta = \begin{cases} \pi - \alpha, & \text{if } (\vec{a} \times \vec{b} > 0), \\ \alpha - \pi, & \text{if } (\vec{a} \times \vec{b} < 0). \end{cases} \quad (4)$$

We define four patterns from simple to complex, as shown in Figure 8. The simple pattern means the trajectory is nearly one directional and has a low curve value. The circle pattern has the same AP as the start and the end point, while the to-and-from pattern means the user shuttles from two APs. If a path has a high curve and HCR values but does not contain circle and to-and-from patterns, it might belong to a complex pattern.

For the TPM, we summarize 12 high-level trajectory features like minimum speed, maximum speed, variance of speed, and the number of large-angle rotations according to HCRs that are introduced by articles [4, 5] for trajectory classification. Besides these, we introduce time sections as our additional features extracted from the SPM. These features are explainable and location independent so that they could avoid location-sensitive drawbacks.

3.4. Classification Layer. Here, we choose traditional classifiers like native Bayes (NB), random forest (RF), logistic regression (LR), gradient boosting decision tree (GBDT), and k-nearest neighbor (KNN) as our classification methods. Moreover, we use one-class SVM (OCSVM) to treat suspects as outliers from normal residents.

We use the undersample method on negative instances to balance the samples of two categories; in detail, we randomly sample negative instances the same as the number of positive instances, train them in a equal magnitude, but test them in an unbalanced dataset, where the ratio of positive instances to negative instances is 1:100. All the methods will repeat 10 times in training and validating processes with a randomly selected dataset and report the averaged results.

4. MST-CNN Framework

Motivating success of embedding techniques in other areas, we now utilize deep neural network methods to classify

```

Input: all the path  $P$  of a certain people
Output: active tuples  $tp = \langle \text{encode\_geo}, \text{time\_slice} \rangle$ 
(1)  $tp \leftarrow \langle \rangle$ 
(2) for every  $O_i$  in  $P$  do
(3)   if  $\text{Geohash}(O_i) \neq \text{Geohash}(O_{i-1})$  then
(4)      $t_1 \leftarrow \text{TimeSlice}(O_i), t_2 \leftarrow \text{TimeSlice}(O_{i-1})$ 
(5)     if  $t_1 = t_2$  then
(6)        $tp \leftarrow (\text{Geohash}(O_i), t_1), (\text{Geohash}(O_{i-1}), t_1)$ 
(7)     if  $t_1 = t_2 + 1$  then
(8)        $tp \leftarrow (\text{Geohash}(O_i), t_1), (\text{Geohash}(O_{i-1}), t_2)$ 
(9)     else
(10)       $tp \leftarrow (\text{Geohash}(O_i), t_1), (\text{Geohash}(O_i), t_1 - 1)$ 
(11) return  $tp$ 

```

ALGORITHM 2: Location-changed model (LCM).

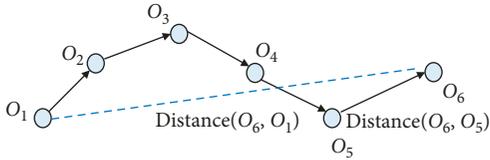


FIGURE 6: Trajectory curve.

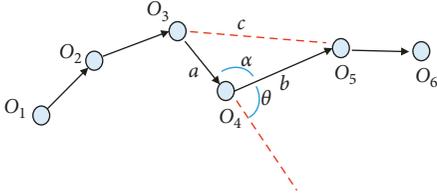


FIGURE 7: Head change rate in a trajectory.

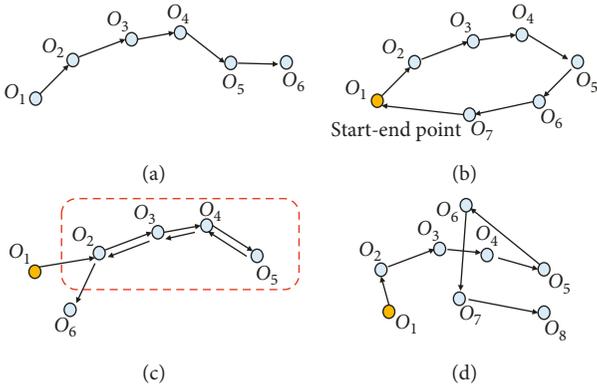


FIGURE 8: Trajectory patterns. (a) Simple pattern. (b) Circle pattern. (c) To-and-from pattern. (d) Complex pattern.

trajectories. After preprocessing moving records by TCA and LCM methods, we embed trajectories in location and time dimensions and generate user behaviors with their related trajectories. Based on the previous work [6], our model multiple spatial-temporal convolutional neural networks (MST-CNNs) have four main steps, as shown in Figure 9. First, we slice trajectories to multiple segments according to a fixed length; second, we embed the segments with location

and time values and concatenate them directly; third, we max pool the vectors of each trajectory and combine them to represent user mobility, which could indicate the aforementioned trajectory features: location information, length and shape of the trajectory, activity time, and so on; Fourth, the vector of user mobility is operated by a neural network comprising two convolutional and max-pooling layers with the ReLU activation function to identify users.

4.1. Structure of Trajectory Embedding. Given a trajectory $T_i^m = (O_1, O_2, \dots, O_n)$, where $O_i \in \nu$ (ν is the Wi-Fi probe), we use PointEmb and TimeEmb functions to demonstrate vectors of T_i^m . According to the word2vec method [7], the input of PointEmb is a large corpus of trajectories, and its output is a vector space where each unique location object O_i has a corresponding vector v_i in an m -dimensional space. In order to illustrate the timestamp feature of the trajectory, We also introduce time information, where a day splits into 24 slices and each slice is a time section:

$$t = [0 \dots 1 \dots 0], O_x(t) \in p\text{th hour } p \in [0, 23], \quad (5)$$

where the j -th entry of t encodes the activity behavior (i.e., the human movements) at $j+1$ -th hour. We further add “positional encodings” by sine and cosine functions of different indexes [8].

4.2. Structure of Personal Encoding. We represent a user u with n trajectories as $v_n^u = [v_1^u, v_2^u, \dots, v_n^u]$, where a single trajectory embedding is concatenated to generate its vector space. A convolutional layer involves a filter $w \in \mathbb{R}^{hk}$ applied to a segment of h moving objects to generate new features. A feature c_i is derived from a segment $v_{i:i+h-1}^u$ by $c_i = f(w \cdot v_{i:i+h-1}^u + b)$, where $b \in \mathbb{R}$ is a bias term and f is a nonlinear function, such as ReLU. For the next step, we introduce a max-pooling layer which is used to get the fixed matrix size after the convolution operation though we get various size inputs and extract the maximum value over the matrix. In order to capture more effective information, we apply various filters (with different segment lengths) to generate different features. Finally, a fully connected layer

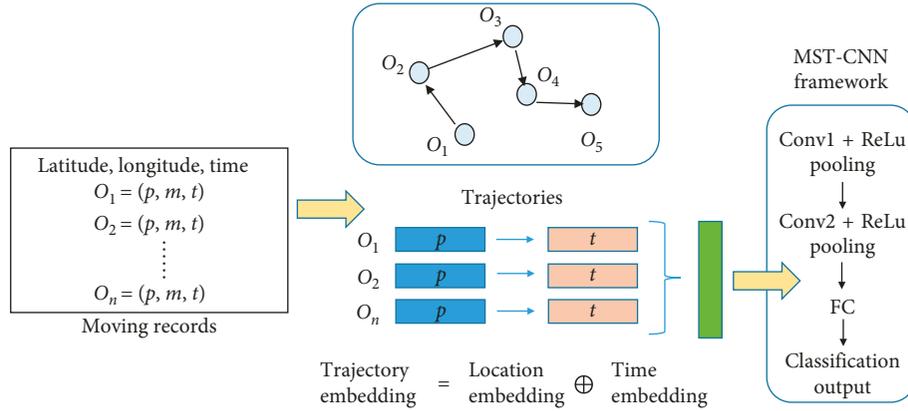


FIGURE 9: Process of the MST-CNN method.

with dropout and softmax outputs is designed to complete the task of user behavior pattern classification.

5. Experimental Results

In this section, we present experimental results with three classification models and a deep learning model. In our experiments, the transit data are collected from the city in East China. The Wi-Fi probe devices are installed around the city and gather nearly 2-3 billion transit records per day from more than 8 million unique MACs. We use the Spark framework to extract three-month records in three months.

- (i) *Platform*. It is a Dell server 64-bit system (16-core CPU, with 2.6 GHz and 32 GB memory), which is used to run classification tasks with various models. The algorithms and models in our paper were implemented by Python 3.
- (ii) *Dataset*. The moving data are gathered from popular Android phones, such as Huawei, Xiaomi, OPPO, Vivo, and Meizu, which accounted for over 75% market in 2017. And then, we need to determine whether a given MAC belongs to a resident or a tourist. In our case, only those users who have at least 15-day transit data in a month would be maintained for our further research. Moreover, we get the suspect list as positive instances from Police Security Department.

5.1. Mobility Movement Perspective. In the above sections, we propose three models to illustrate movement features. Before we do a comparison, we take a look at the mobility movement perspective of both normal residents and suspects. It is common that most human activities during weekends (i.e., leisure activities and family shopping) do not have such strict time and location as the ones during weekdays (i.e., going to the workplace), so it is highly possible that their regular patterns are different between these times.

From Table 2, we could see normal residents prefer to travel farther and longer trips with a higher speed (like by

car), while suspects like to move locally with a lower speed (like by walk). From the perspective of time, the trajectory length, curve, and duration values at night are larger than those at other times.

5.2. Result Comparison. We use *precision*, *recall*, and *F measure* computed with test data to evaluate the effectiveness of three models. *Precision* is the number of correct positive results divided by the number of all positive results returned by the classifier, and *recall* is the number of correct positive results divided by the number of all relevant samples. Then, *F measure* is defined as $F_\beta = (1 + \beta^2) \times (\text{precision} \times \text{recall}) / (\beta^2 \times \text{precision} + \text{recall})$.

Since the dataset in actual is unbalanced and the precision is very low for all the models, they cannot be the direct guide to our business. We use F_3 measure, which weighs recall higher than precision.

In order to make the comparison fair, we use 10-fold cross-validation by dividing the dataset into 70% for training and 30% for testing.

We can see the experiment results shown in Table 3 and do further analysis of the results which confirms our research motivation. The static POI model (SPM) has the lowest performance in all the methods, not only the traditional machine learning methods but also the deep learning method. Generally, the improved LCM is 2-3 times higher in precision than the SPM and is nearly 10 percent better in the recall. The performance of the trajectory pattern model (TPM) is better than that of the SPM, but in some cases, it is inferior to that of the LCM. Furthermore, if we combine the LCM and TPM together, which will provide both spatiotemporal feature and trajectory feature, the performance promotes significantly both precision and recall with all the classifiers. This observation shows that dynamic features play more important roles in the suspect-detection task. Although the positive data are very sparse (less than 1%), the LCM + TPM could get 5.1% precision with 97% recall by GBDT. Inspired by the classification results, we believe that the extracted features play a vital role; thus, we further use these features combined with time-embedding vectors and location-

TABLE 2: Movement perspective of different groups of people.

Features	Morning (6~9 a.m.)		Afternoon (16~19 p.m.)		Midnight (1~5 a.m.)	
	Normal	Suspect	Normal	Suspect	Normal	Suspect
Activity scope	4.33	3.86	4.35	3.76	3.76	3.70
Trajectory length	24.90	7.58	24.43	6.87	44.59	6.46
Trajectory curve	23.49	6.65	19.03	6.67	37.28	5.22
Speed	13.67	16.41	13.28	16.50	13.36	20.41
Velocity change rate	0.22	0.20	0.18	0.22	0.15	0.19
Movement pattern	0.16	0.05	0.16	0.04	0.05	0.03
Trajectory duration	2.01	0.54	1.93	0.50	3.32	0.39
Head change rate	0.44	0.44	0.43	0.51	0.38	0.47

TABLE 3: Experimental results with different classification methods.

Model	Methods	Accuracy	Precision	Recall	F_3 measure
RF	SPM	0.818	0.009	0.602	0.079
	LCM	0.867	0.028	0.786	0.212
	TPM	0.912	0.021	0.694	0.165
	LCM + TPM	0.915	0.042	0.96	0.301
NB	SPM	0.852	0.007	0.714	0.064
	LCM	0.861	0.018	0.726	0.147
	TPM	0.923	0.015	0.857	0.114
	LCM + TPM	0.935	0.014	0.818	0.121
GBDT	SPM	0.807	0.012	0.654	0.103
	LCM	0.844	0.029	0.785	0.218
	TPM	0.956	0.022	0.703	0.172
	LCM + TPM	0.962	0.051	0.972	0.346
LR	SPM	0.844	0.009	0.651	0.08
	LCM	0.851	0.013	0.673	0.111
	TPM	0.892	0.018	0.664	0.145
	LCM + TPM	0.906	0.019	0.886	0.159
KNN	SPM	0.837	0.014	0.71	0.119
	LCM	0.852	0.034	0.71	0.238
	TPM	0.869	0.015	0.69	0.125
	LCM + TPM	0.950	0.035	0.86	0.256
OSVM	SPM	0.907	0.018	0.912	0.153
	LCM	0.912	0.019	0.995	0.162
	TPM	0.909	0.018	0.932	0.154
	LCM + TPM	0.921	0.017	0.930	0.146
Deep learning	MST-CNN	0.981	0.191	0.650	0.524

embedding vectors fed into the neural network, and it turns out that using deep learning methods helps obtain great performance. The precision of this method is 0.191 which far exceeds that of other methods, the same as the result of accuracy and F_3 measure.

In order to check the effectiveness and stability of models, we modify the size of training data and measure the F_3 measure with classification methods mentioned above. Figure 10 states that the SPM and LCM do not have stable performance, and F measure is somehow lower when the training data are growing. But the TPM and LCM + TPM have better and stable performance. The more the training data, the better the classification result.

5.3. Behavior Visualization and Insights. Although the LCM + TPM could improve the precision and recall of classification as Figure 10 states and the movement features

of normal residents and suspects are obviously different in Table 2, we still need some visualization tools on the map to help us understand movement patterns better.

Figure 11 gives an example of a normal people transit pattern on the map. The blue line presents the transit path of a normal resident, while the red line belongs to a suspect. As expected, the blue line is larger, longer, and more complicated than the red line.

Figure 12 illustrates the instant speed and average speed of two groups. As visualized, residents prefer to choose a faster transit tool after work. They might change the transit mode during the travel path, so the instant speed is changed. In contrast, the suspect does not like to go far away and likes to stroll locally. Since many suspects are unemployed, they choose to move by walk in the rush hour.

Through the above analysis, we can find that the normal residents and the suspects do have significant differences, which is why we can use the feature extraction on the

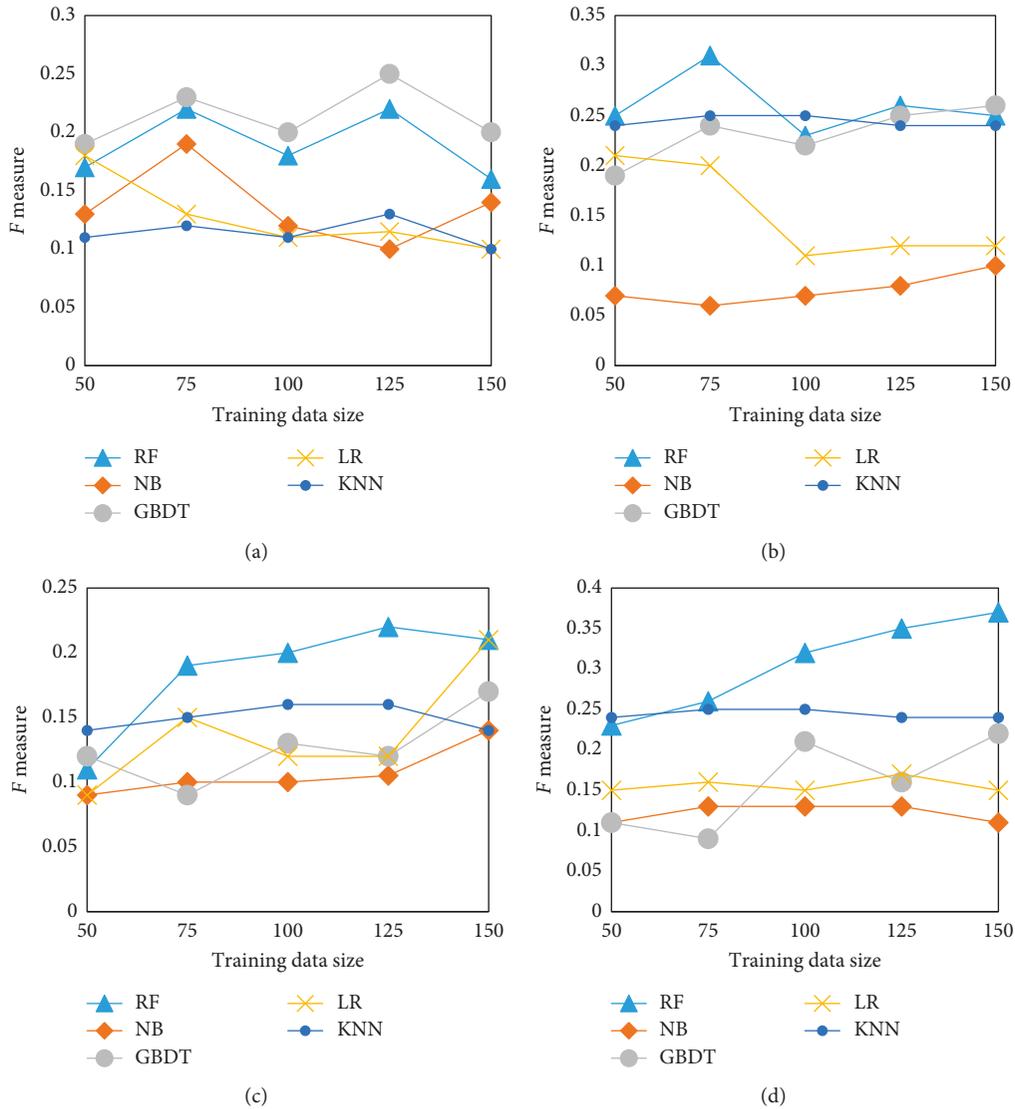


FIGURE 10: Performance change with training data size: F measure of the (a) SPM; (b) LCM; (c) TPM; (d) LCM + TPM.

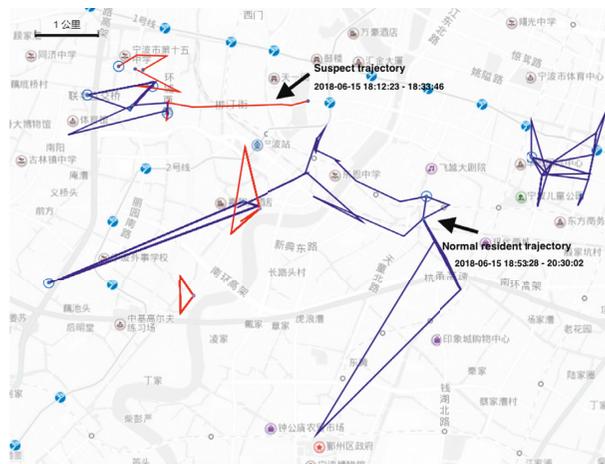


FIGURE 11: Trajectories of two groups of people on the map.

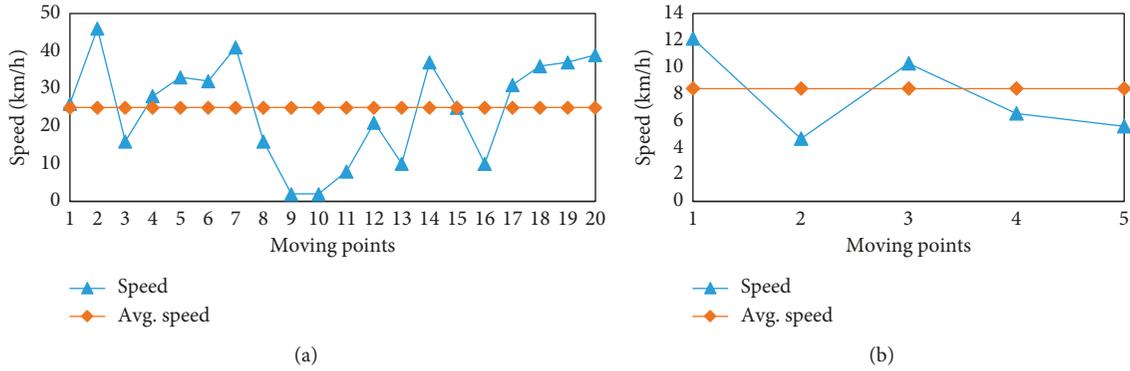


FIGURE 12: Speed mode of the trajectory on the map. Trajectory speed of the (a) resident and (b) suspect.

trajectory to identify whether the user is a suspect or not. According to our experimental results, the model based on deep learning performs the best compared to other methods; one of the reasons is because the neural network can extract features of the target better. However, neural networks are often uninterpretable and data sensitive. Even if they obtained great performance in this case, they may do badly elsewhere; thus, it is quite important to analyze a specific case and extract vital features from the data, like in our study, and the extra features extracted by the proposed models also play an indispensable role in the result of MST-CNN.

6. Related Works

The related works fall into two categories: movement pattern mining (MPM) and behavior understanding.

6.1. Movement Pattern Mining. Traditional machine learning methods are applied to extract patterns for classification. Gong et al. [9] use GPS data to detect five travel models (walk, car, bus, subway, and commuter rail) in New York. Pinelli proposes an extension of the sequential pattern-mining paradigm to analyze the trajectories of moving objects [10]. REMO (Relative MOTion) [11] is based on the traditional cartographic approach of comparing snapshots and develops a comparison method based on motion parameters to reveal the movement patterns. Li et al. [12] present an efficient probabilistic model to analyze GPS snippet data. MPM could also be utilized on other study topics such as bus route planning by taxi traces [13], finding the frequent path of passengers [14], or discovering and explaining movement patterns of a set of moving objects [15].

6.2. Behavior Understanding. Understanding trajectories of users would improve the effectiveness of machine learning models. For example, Du et al. analyze passengers' transit behaviors from subway transit records and discover the pickpocket suspects by abnormality detection [2]. Yuan et al. design a framework that learns the context of different functional regions in a city [4]. The regularity of

historical trajectories could also be considered as effective features [5, 16]. Abul proposes a W4M (wait for me) method, which uses the edit distance to measure the similarity of different paths. Considering the mobility similarity between user groups, Zhang et al. [17] propose the GMove modeling method to share significant movement regularity.

Trajectory classification benefits a wide variety of deep learning methods. ST-ResNet [18] is designed to forecast the flow of the crowd. The DeepMove [19] model predicts human mobility with the attentional recurrent network, while HST-LSTM [20] captures location prediction by spatial-temporal LSTM. Motivated by these research studies, our previous work also based on deep learning methods classifies suspects, but without feature insight analysis [6].

7. Conclusion

In this paper, we focus on the suspect identification problem via large-scale trajectory data. We propose effective LCM and TPM methods to extract transit features of moving behaviors like traveling speed, activity scope, path length, and trajectory shape, which could avoid the location-sensitive problem. We also design a deep learning method for an end-to-end classification skipping feature selection step. Moreover, we describe suspects' activities by statistical analysis and set up a real-life dataset for training and validation. Experimental results show our model has a better performance in both precision and recall and the effectiveness of trajectory-based features is stable in various sizes of datasets.

Data Availability

The transit record data used to support the findings of this study were supplied by some surveillance systems of Public Security Department in China. Since data would reveal personal activities and the size of data is increased by additional sensors during recent days, these data cannot be made freely available. We are glad to supply part of data after removing the personal information and unique IDs for research. Requests for access to these data should be made to Canghong Jin (e-mail: jinch@zucc.edu.cn).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China (No. U1509219) and Science & Technology Development Project of Hangzhou, China (No. 20162013A08).

References

- [1] Police Urges Google to Provide User Location Information as “For Suspect Search” https://gigazine.net/gsc_news/en/20180321-police-quietly-turn-to-google/.
- [2] B. Du, C. Liu, W. Zhou et al., “Catch me if you can: detecting pickpocket suspects from large-scale transit records,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 87–96, ACM, San Francisco, CA, USA, August 2016.
- [3] N. Council, *Data for Understanding Our Nation’s Travel: National Household Travel Survey Conference*, Transportation Research E-Circular, Washington, DC, USA, 2005.
- [4] J. Yuan, Y. Zheng, and X. Xie, “Discovering regions of different functions in a city using human mobility and POIs,” in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining-KDD’12*, pp. 186–194, Beijing, China, August 2012.
- [5] Q. Li, Y. Zheng, X. Xing et al., “Mining user similarity based on location history,” in *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Irvine, CA, USA, November 2008.
- [6] C. Jin, H. Liang, D. Chen, Z. Lin, and M. Wu, “Identifying mobility of drug addicts with multilevel spatial-temporal convolutional neural network,” in *Advances in Knowledge Discovery and Data Mining, PAKDD*, pp. 477–488, 2019.
- [7] Y. Kim, “Convolutional neural networks for sentence classification,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing EMNLP*, pp. 1746–1751, Doha, Qatar, October 2014.
- [8] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” in *Proceedings of the 2017 Conference on Neural Information Processing Systems*, pp. 5998–6008, Long Beach, CA, USA, December 2017.
- [9] H. Gong, C. Chen, E. Bialostozky, and C. T. Lawson, “A GPS/GIS method for travel mode detection in New York City,” *Computers, Environment and Urban Systems*, vol. 36, no. 2, pp. 131–139, 2012.
- [10] F. Pinelli, F. Pinelli, F. Pinelli et al., “Trajectory pattern mining,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 330–339, ACM, San Jose, CA, USA, August 2007.
- [11] P. Laube and S. Imfeld, “Analyzing relative motion within groups of trackable moving point objects,” in *Proceedings of the Second International Conference on Geographic Information Science, GIScience 2002*, pp. 132–144, DBLP, Boulder, CO, USA, September 2002.
- [12] M. Li, A. Ahmed, and A. J. Smola, “Inferring movement trajectories from GPS snippets,” in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pp. 325–334, ACM, Shanghai, China, February 2015.
- [13] C. Chen, D. Zhang, Z.-H. Zhou, N. Li, T. Atmaca, and S. Li, “B-planner: night bus route planning using large-scale taxi gps traces,” in *Proceedings of the 2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 225–233, San Diego, CA, USA, March 2013.
- [14] W. Luo, H. Tan, L. Chen, and L. M. Ni, “Finding time period-based most frequent path in big trajectory data,” in *Proceedings of the 2013 International Conference on Management of Data-SIGMOD’13*, pp. 713–724, New York, NY, USA, June 2013.
- [15] T. L. Coelho da Silva, J. A. F. de Macêdo, and M. A. Casanova, “Discovering frequent mobility patterns on moving object data,” in *Proceedings of the Third ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems-MobiGIS’14*, pp. 60–67, Dallas, TX, USA, November 2014.
- [16] J. C. Ying, H. C. Lu, W. C. Lee et al., “Mining user similarity from semantic trajectories,” in *Proceedings of the ACM SIGSPATIAL International Workshop on Location Based Social Networks*, pp. 19–26, ACM, San Jose, CA, USA, November 2010.
- [17] C. Zhang, K. Zhang, Q. Yuan et al., “GMove: group-level mobility modeling using geo-tagged social media,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1305–1314, ACM, San Francisco, CA, USA, August 2016.
- [18] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence AAAI*, pp. 1655–1661, San Francisco, CA, USA, February 2017.
- [19] J. Feng, Y. Li, C. Zhang et al., “DeepMove: predicting human mobility with attentional recurrent networks,” in *Proceedings of the 2018 World Wide Web Conference on World Wide Web WWW’18*, pp. 1469–1468, Lyon, France, April 2018.
- [20] D. Kong and F. Wu, “HST-LSTM: a hierarchical spatial-temporal long-short term memory network for location prediction,” in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI*, pp. 2341–2347, Stockholm, Sweden, July 2018.



Hindawi

Submit your manuscripts at
www.hindawi.com

