Sensors and Materials, Vol. 35, No. 12 (2023) 4085–4104
MYU Tokyo

S & M 3469

Fuzzy Spatiotemporal Representation Model for Human Trajectory Classification

Lifeng Chen,[1] Canghong Jin,[1*] Hao Wu,[2] Jiafeng Zhao,[3] and Jianghong Wu[4**]

[1]Hangzhou City University, 51th Huzhou Street, Hangzhou, Zhejiang 310015
[2]Macau University of Science and Technology
[3]Zhejiang University of Technology
[4]Zhejiang Key Laboratory of Social Security Big Data

(Received July 14, 2023; accepted November 30, 2023)

Keywords: trajectory encode, behavior representation, trajectory classification, spatiotemporal fuzzification

Effective trajectory selection and classification are pivotal in user tracking systems utilizing spatiotemporal data collected from city sensors. However, the inherent limitations in sensor technologies and data collection point distributions often result in low-quality spatiotemporal data. Real-life trajectory classification encounters challenges due to the following: (1) high-order and sparse activity data encompassing both temporal and spatial contexts, and (2) inherent vagueness in the semantics of visited locations, making it difficult to represent behavioral intentions. Traditional statistics-based or trajectory-based feature approaches prove ineffective with non-discriminate features. In response to these challenges, we introduce a novel classification method that integrates fuzzy spatiotemporal features and crowd habit features. This approach involves feature extraction using the Time-Geo Hash (TGH) and User Transit Pattern and Similarity (UTPS) models, followed by the training of a machine learning classification model. On the basis of the performance indicators of classification models, we identify two classification algorithms, incorporate the Bagging algorithm from ensemble learning to enhance the UTPS classification model, and combine the TGH and UTPS models through specified rules. Extensive experiments demonstrate that our proposed model significantly outperforms other classification baselines when applied to a labeled real-life dataset, emphasizing its effectiveness in handling noisy and challenging spatiotemporal data for trajectory classification in user tracking systems.

1. Introduction

The widespread usage of the mobile Internet has resulted in a heavy reliance on mobile devices to access online services. Concurrently, the massive data collected by these devices capture individuals' behavioral features, particularly their spatiotemporal patterns, which have attracted significant attention from researchers. It is a consistent and significant direction to

*Corresponding author: e-mail: jinch@hzcu.edu.cn
**Corresponding author: e-mail: 13867180080@139.com
https://doi.org/10.18494/SAM4590

ISSN 0914-4935

study the use of spatiotemporal information for identifying and classifying different behavioral groups.[1]

The collection of movement behaviors can be achieved comprehensively by utilizing location information from cell phones. The movement trajectory data encompasses specific time and location information, essentially, spatiotemporal data. By leveraging spatiotemporal information, it becomes possible to reconstruct individuals' activity trajectories, enabling the analysis of their habitual characteristics and behavior patterns.[2–4] A crucial application scenario of spatiotemporal data involves classifying individuals based on spatiotemporal information. Extracting discriminated features that strongly correlate with specific application scenarios is very useful for machine learning. For example, Du *et al.*[5] utilized user check-in data from social media during morning rush hours, working hours, evening rush hours, and nonworking leisure hours as features, thereby acquiring user movement behaviors from datasets. By employing the *k*-means clustering method and *k*-nearest neighbor algorithm, citizens were successfully classified on the basis of the above features, facilitating the identification of additional personal details such as workplace, residence, and occupation. Furthermore, researchers have investigated the regularity of individuals' activity trajectories. Song and coworkers [6,7] highlighted that people's activities can be predicted, with a predictability rate of up to 93%. De Montjoye *et al.*[8] proved the uniqueness of individuals' activities. Additionally, it can be concluded that the activity trajectories of individuals are related to their social connections, as observed in Ref. 9: the closer the relationship between individuals and their social connections, the greater the similarity observed in their activity trajectories.

Despite the significant benefits of these methods in classifying or predicting human behaviors, they present some challenges to our problem. The selection of trajectories from the vast spatiotemporal data collected by sensors in a city encounters limitations. Owing to limitations in collection technologies and the distribution of collection points, the spatiotemporal data is frequently characterized by sparsity, positional offset, and high feature dimension. Moreover, the sparsity of data hampers the accurate reflection of activity trajectories, whereas the high feature dimension introduces computational complexity and can potentially impair the performance of conventional classification models, namely, the curse of dimensionality.[10] Consequently, conventional classification models struggle to attain satisfactory results in this context.

To address the challenges associated with sparse and diverse trajectory data, we propose a method of extracting crowd habit features on the basis of fuzzy spatiotemporal data. The proposed method aims to improve classification performance by integrating it with classification models. The key steps and contributions are as follows.

- Regarding high spatiotemporal dimensions in data, we come up with the Time-Geo Hash (TGH) model. The TGH model effectively handles time information by processing it in fragments and encoding spatial location information in a vague manner. Additionally, the TGH model maps adjacent acquisition time points to the same time slice, thereby reducing the number of time dimensions. Furthermore, applying the hash algorithm to the location information mapping of the collected data significantly reduces the number of spatial dimensions.

- The User Transit Pattern and Similarity (UTPS) model is developed to extract user habits. It involves the calculation of spatiotemporal information collected through the media access control (MAC) address, which can be regarded as the identity of the mobile device during various work and rest periods, as well as the evaluation of the similarity of daily activities for each MAC. The model depicts the intensity and regularity of daily activities in different regions in various time periods of each MAC. Additionally, the Bagging algorithm of ensemble learning is introduced to imporve the UTPS model.
- Eventually, the TGH and UTPS models are synergistically combined for comprehensive decision-making. The experimental results show that the combined model considerably improves the classification accuracy compared with a single model. *Lift* value calculation results indicate that the proposed model can better classify and predict people with diverse behaviors.

2. Related Work

2.1 Ensemble learning

Ensemble learning, which originated from the concepts of strongly and weakly learnable concepts,[10] has emerged as a fundamental technique in machine learning. It has proven to be instrumental in improving the generalization ability and prediction accuracy of classifiers.[11] Integrated learning can be observed in both narrow and broad senses. In the narrow sense, multiple subsets are randomly selected from the training set, and the same classification algorithm is applied to each subset to enhance the generalization ability of each classifier. On the other hand, in the broad sense, the same problem is tackled using multiple learners. The ensemble learning process mainly involves three steps: generating training subsets, training base classifiers, and integrating the results obtained from these classifiers. Bagging[12] and Boosting are representatives and most commonly used in ensemble learning methods.[13]

2.2 Similarity calculation of spatiotemporal information

According to the spatiotemporal information used in calculating similarity, research on the similarity of spatiotemporal information can be divided into three categories.

(1) **Spatial similarity** focuses solely on spatial information and does not consider temporal aspects. Research studies on this primarily explore the geometric shapes of spatiotemporal trajectories using distance metrics such as Euclidean distance,[14,15] longest common subsequence (LCSS),[16] edit distance on real sequence (EDR),[17] and graph structure similarity[18] to measure the similarity between trajectories.

(2) **Time similarity** concentrates on analyzing the similarity based on time series data. For example, the Fast search method for dynamic time warping (DTW)[19] is used to calculate the distance between two time series.

(3) **Spatiotemporal similarity** considers both spatial and temporal information, such as by implementing *k*-most-similar-trajectory (K-MST) queries using data structures similar to R-trees,[20] utilizing low-resolution trajectories to compose sets of crowd classification rules

(FCRs),[21] or employing the top-bottom clustering algorithm for small crowd classification based on offline crowd trajectories.[22]

The representation form of spatiotemporal information in the similarity calculation process can be divided into two categories: multidimensional vector and string forms. The multidimensional vector form requires more computational resources but provides a higher accuracy.[23] Therefore, in this study, the multidimensional vector form is utilized.

3. Problem Formulation

Here, we introduce several basic concepts and provide a formal definition of the user identification problem.

Definition 1 (Moving Point): The moving point is represented by $O = (p, m, t)$, where p represents the location information, including latitude, longitude, and the name of the monitoring point, m refers to the MAC information, and t indicates the time information.

Definition 2 (Path): Given a set of moving points $<O>$ and a specific MAC address a, a path associated with a can be expressed as $P = \{O_1, O_2, \ldots, O_n\}$, where $\forall O_i(m) = a$, and for $i < j$, $O_i(t) < O_j(t)$.

Definition 3 (Person of Interest): MAC addresses can be classified into two types: **Person of Interest** and **General Public.** The former consists of MAC addresses provided by the public safety department of a city, representing individuals who are of specific security or investigative interest. These addresses are assigned on the basis of criteria such as suspected criminal activity, surveillance targets, and involvement in ongoing investigations. The focus is on monitoring the movements and activities of these individuals for public safety and security purposes. The latter category includes MAC addresses associated with the general population.

Definition 4 (Problem): The Person of Interest Identification problem is defined as judging whether or not a MAC address is a POI on Path *P*. The model is improved to obtain classification as precise as possible.

4. Methodology

4.1 Overall framework

As shown in Fig. 1, the overall framework is constructed according to the following steps: (1) data preprocessing, (2) spatiotemporal information coding and user behavioral modeling, (3) Bagging algorithm, and (4) outputting the final prediction result.

The data on the far left in the figure represents the quintuple obtained after data cleaning. Through subsequent feature extraction and preprocessing, a dataset is constructed to train the classification model, that is, the training set. On the basis of the MAC address list of persons of interest provided by the public security department, classified labels can be determined in a training set, or in other words, whether they belong to a specific group of people.

Note that the original location information is organized in the order of collection point and

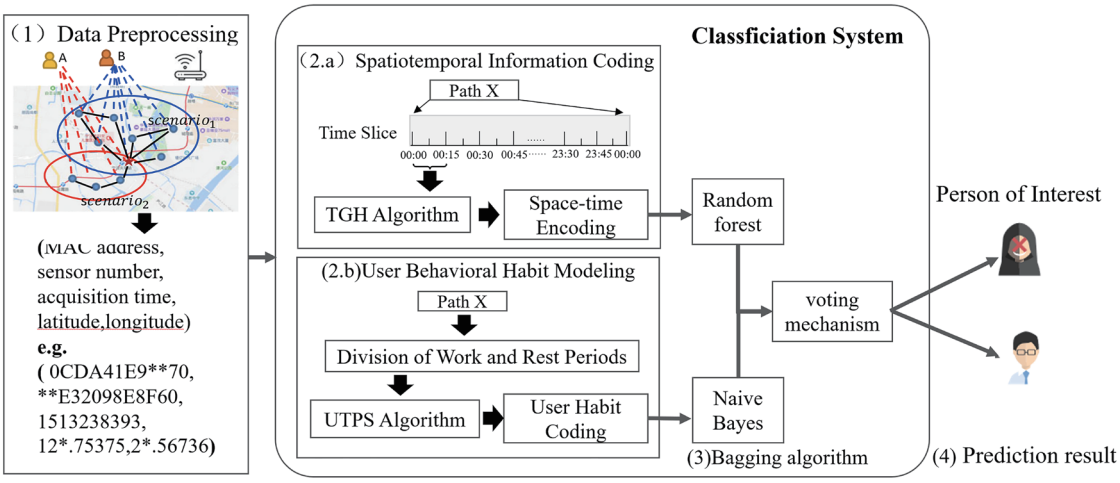

Fig. 1. (Color online) Overview of fuzzy trajectory classification system.

time. The data quintuple after data cleaning indicates that the MAC address serves as the unit, and features are extracted from multiple activity records corresponding to a single MAC address and aggregated into a sample. Granularity is no longer a single record but the MAC address of mobile devices.

As depicted in the middle of Fig. 1, our approach aims to address dimensionality reduction and compensate for data imperfections. To achieve this, we proposed two feature extraction algorithms, namely, TGH and UTPS, which offer distinct perspectives for generating training sets. These algorithms help overcome limitations in the data and improve its overall quality.

Subsequently, the determined classification algorithms are employed to train and combine these feature extraction approaches into a robust classifier. To obtain the final prediction result, we employ a voting mechanism within ensemble learning. By synthesizing the outputs of the two feature extraction algorithms, we achieve an enhanced predictive capability that leverages the strengths of both approaches.

4.2 Data sources and data preprocessing

4.2.1 Data sources

The spatiotemporal data in our study were mainly collected via public safety sensors in a city. The data model can be represented by the following quintuple:

(*MAC address, sensor number, acquisition time, sensor's latitude and longitude*).

The quintuple can describe moving tracks of a single MAC in various time and space domains.

4.2.2 Data preprocessing

Since the collected data are affected by duplication and incompletion, it is necessary to preprocess abnormal data. Moreover, we should also deal with other problems during research, such as the uniform conversion of particular time formats in the data reported by some sensors, remove the hyphens (-) in data, and convert decimal MAC addresses into hexadecimal ones. After that, other smart devices such as smart air conditioners, smart sockets, and smart home products should be identified and discarded.

4.3 Spatiotemporal information coding

4.3.1 TGH algorithm

In this section, we encode a given path X in time and space to reduce its dimensionality. First, we divide the continuous time information by the TGH algorithm and map adjacent acquisition time points to the same time slice (*ts*). Then, we use the UTPS algorithm to map the collected data's location information and encode the geolocation region into a hash value *lh* (location hash). The pseudocode is described in Algorithm 1.

4.3.2 TGH algorithm details

Time slice:

The TGH algorithm divides time into four slices, specifically, a quarter of an hour as a unit, and divides one hour into four slices. Thereby, there are 96 slices in a day. They are numbered from 1 to 96, and each is called a time slice. In Fig. 1, the time point on the right side of each time slice is taken from an open interval. For example, the time period corresponding to time slice number 1 is [00:00, 00:15] and so on.

Through experimental comparison, 15 min was selected as the time slice granularity. When every 15 min is taken as a time slice, moderate time dimensions are produced with good activity discrimination.

Geohash location encoding:

The latitude and longitude are encoded into a alphanumeric string by Geohash. The coded string represents a rectangular region of Earth. As shown in Table 1, the greater precision obtained with longer strings.

Algorithm 1: Time-Geo Hash (TGH)

Input: given path P
Output: Encoded path feature queue <*PF*> // Path Features
<*PF*> initialized to empty
for O_i in P **do**
    $ts \leftarrow \text{TimeSlice}(O_i)$, $lh \leftarrow \text{GeoHash}(O_i)$
    if *ts*, *lh* ∈ <*PF*> **then**
        continue;
    else
        <*PF*> ← *ts*, *lh*
return <*PF*>

Table 1
Correspondence between Geohash coding length and error.

Code length	Error (km)
1	±2500
2	630
3	78
4	20
5	2.4
6	0.61
7	0.076
8	0.019

Considering the division of urban functional blocks and the dimension quantity of the entire city, latitude and longitude coordinates are converted into 5-bit Geohash codes. This conversion enables a more precise representation of geographic locations. As seen in the table, each Geohash code roughly corresponds to the length of 2.4 km of the rectangular region.

Space-time encoding:

A single MAC address yields path information, and after encoding through the TGH model, it contains a total of 486 features as below.

(*label, mac, time-features, geo-features*)

For a specific MAC address, the labels are represented in the first column, indicating whether the owner of the mobile device associated with the MAC address is a follower. A label value 1 denotes "yes," whereas 0 denotes "no." The second column denotes the unique MAC address. The third column consists of a series of values representing the number of times the MAC address is collected in each time slice. This sequence is determined to be 96 dimensions, as established previously. Following that, a collection of 5-digit Geohash codes represents the frequency of data collection for the MAC address in each corresponding geographical area. As determined earlier, there are a total of 388 codes.

4.4 User behavioral habit model

4.4.1 UTPS algorithm

The UTPS model is proposed for each MAC address in different work and rest time periods during the working day, and each MAC address corresponds to similarities in the daily activities of the mobile terminal holder.

As shown in Algorithm 2, UTPS is mainly used to describe the degree and similarity of activities of users in different time periods and regions. It can be used to compensate for deficiencies such as missing data collection. In addition, the similarity of daily behavior is incorporated into the statistics to extract features from a new perspective.

Algorithm 2: User Transit Pattern and Similarity (UTPS)
Input: given <*P*> within a certain time period of a MAC, monitoring points in key areas <*KAP*>
Output: Habit features <*HF*> // habit features
sum(*f*) = 0, $f \in$ {*record*, *ts*, *lc*, *kap*} // total number of distinct eigenvalues
for *P* **in** <*P*> **do**
Divide trajectory *P* into time segments *p*1 to *p*4 by time, denoted by P_i
for O_i **in** <*P*> **do**
sum(*record*), *sum*(*record*, P_i) ← #O_j // Calculate number of records and total number of records in the top-10 most frequent areas, which is similar for other data
if $O_j(p) \in$ <*KAP*> **then**
sum(*kap*), sum(*kap*, P_i) ← #O_j
Calculate proportion of each feature in each time period hf, <*HF*> ← *hf*
Calculate user behavior similarity, see Sect. 4.4.2
return <*HF*>

4.4.2 UTPS algorithm details

Proportion of spatiotemporal information:

Inspired by some research,[5–8,24] the UTPS algorithm defines the time period division table as below, in accordance with work and rest habits of office workers in work days.

Given the inherent disparities in transit behaviors between work days and holidays, the UTPS algorithm is purposefully designed to prioritize the analysis of data collected exclusively on work days. This approach incorporates the invaluable expertise and experience of the public security department, which serves to comprehensively consider the distinct patterns and unpredictable activities characteristic of nonworking days.

Similarity information:

The UTPS algorithm uses the TGH algorithm to convert all quintuples of data collected daily for each MAC address into a sample piece of data, such as

$$(S_1, S_2, \ldots, S_{dim}, g_1, g_2, \ldots, g_{dim}),$$

where S_{dim} indicates the dimensions of *time features* and g_{dim} the dimensions of *geo-features*. In a specified time period, the number of days on which the activity record of each MAC address is collected corresponds to the number of multidimensional vectors.

Then, we set the dimension of the multidimensional vectors as $v_{dim} = s_{dim} + g_{dim}$ and define all multidimensional vectors as

$$v_i = \left(s_1, s_2, \ldots, s_{s_{dim}}, g_1, g_2, \ldots, g_{g_{dim}}\right),$$

where $i = 1, 2, \ldots, n$ and $v_i[j]$ represents the j-th element of v_i, $j = 1, 2, \ldots, v_{dim}$.

The similarity of space-time vectors of multiple days should be calculated to determine whether the user behavior is regular. The process can be divided into three steps.

Step 1: Combine v_i into a multidimensional vector mix with the same dimensions.

If at least one of the values of the first dimension in v_i is 1, the first dimension of the mix will be 1. Otherwise, it is 0. All subsequent dimensions are processed in return in the same way,

obtaining a multidimensional vector mix of the same dimension. The mix can be viewed as a synthesis of v_i.

Step 2: Calculate similarity between v_i and mix.

The Jaccard index is introduced to calculate the similarity of two multidimensional vectors as

$$sim(i,j) = \frac{q}{q+r+s}. \tag{1}$$

Among them, i and j are two multidimensional vectors. The value of each dimension is 0 or 1. q represents the number of dimensions when the same dimensions of i and j are both 1. r represents the number of dimensions when the same dimension of former values is 1 and that of latter values is 0. s represents the number of dimensions when the same dimension of former values is 0 and that of latter values is 1.

Equation (1) is used to compute the similarity between v_i and *mix*. The denominator clearly represents the count of dimensions in a mix with a value of 1, whereas the numerator signifies the presence of these dimensions in v_i, that is,

$$sim(v_i, mix) = \frac{\sum_{j=1}^{v_{dim}} v_i[j]}{\sum_{j=1}^{v_{dim}} mix[j]}. (i = 1,2,\ldots,n) \tag{2}$$

Step 3: Determine the overall similarity of all multidimensional vectors by computing the average of all similarities in Step 2.

It is easy to calculate the overall similarity via Eq. (3).

$$sim_{total} = \frac{\sum_{i=1}^{n} sim(v_i, mix)}{n} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{v_{dim}} v_i[j]}{n \cdot \sum_{j=1}^{v_{dim}} mix[j]}, (i = 1,2,\ldots,n) \tag{3}$$

When the number of days is 1, indicating that there is only a single multidimensional vector, the *mix* is identical to that vector, resulting in an overall similarity score of 1. Similarly, if all multidimensional vectors are identical, the *mix* will also be identical to these vectors, yielding an overall similarity score of 1.

User habit coding:

The UTPS algorithm generates a dataset comprising sample rows that correspond to individual MAC addresses. This dataset encompasses a total of 15 distinct features as below.

(*label, mac, proportion of p_i records, proportion of p_i regions, similarity of daily activities*)

The proportion of p_i records with i = 1, 2, 3, and 4 is defined in Table 2. It represents people's activities in different time periods. The proportion of p_i regions with i = 1, 2, 3, …, 10, represents how active the person is in the top-10 most active areas.

4.5 Model improvements

To enhance the classification model's performance, we leverage the Bootstrap aggregating algorithm within the realm of ensemble learning. This technique combines multiple weak classifiers, each trained by a specific algorithm, to form a robust classifier that delivers the ultimate prediction.

We focus on enhancing the classification model in four key areas: (1) reducing feature dimensionality, (2) selecting appropriate algorithms, (3) employing Bagging ensemble techniques, and (4) enhancing comprehensive decision-making. As we have already explored feature dimensionality reduction, the subsequent sections will delve into the latter three aspects in detail.

4.5.1 Algorithm selection

Performance indicator of classification model*:*

Accuracy (*ACC*) is calculated using Eq. (4).

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

In cases of imbalanced data, the accuracy metric may not adequately capture the overall performance of the classification model. Therefore, we introduce other performance indicators, namely, *Precision* (precision), *Recall* (recall rate), and *F*1, as

$$Precision = \frac{TP}{TP + FP}, \tag{5}$$

$$Recall = \frac{TP}{TP + FN}, \tag{6}$$

Table 2
Division of work and rest periods.

No.	Time period	Time range
*p*1	Rest period	[21:00, 06:00)
*p*2	Commute time (work)	[06:00, 10:00)
*p*3	Working hours	[10:00, 17:00)
*p*4	Commute time (get off)	[17:00, 21:00]

$$F1 = \frac{2}{1/Precision + 1/Recall}. \quad (7)$$

Within these metrics, *Precision* signifies the fraction of samples accurately predicted as the target category out of all the samples predicted as such, whereas *Recall* denotes the portion of correctly predicted samples among those belonging to the target category. These two metrics offer distinct viewpoints on the classification model's performance. A higher *Precision* implies that the classification model seldom misclassifies nontarget samples as the target category, whereas a higher *Recall* suggests that the classification model rarely misclassifies the target category as nontarget. The *F*1 score serves as a combined metric that considers both *Precision* and *Recall*.

TGH classification model algorithm selection:

Various machine learning classification algorithms are used to train models on the TGH training set. On the basis of the performance metrics mentioned earlier, we have opted for a random forest classifier to classify the feature datasets generated by the TGH algorithm (refer to the experimental section for a comparison of performance indicators of various classification algorithms).

Random forest is a classifier composed of decision trees. Each decision tree judges a new sample to be classified. The most frequent classification category in the result is taken as the final classification of the sample on the basis of the classification results of each decision tree. This process is called "Votes".

Random forest performs better in classifying many datasets because multiple decision trees vote. It can process data with many features and does not select features. The disadvantage is that building a forest takes up more memory, and overfitting will occur on some datasets that include diverse categories of features.

Random forest performs the best among many classification algorithms to classify feature datasets extracted by the TGH algorithm. This is probably related to its suitability for multi-dimensional feature data and applying multiple decision trees for comprehensive voting.

UTPS classification model algorithm selection:

Classification models are trained by different machine learning classification algorithms on the UTPS training set. On the basis of the above performance indicators, Naive Bayes is selected to classify feature datasets extracted by the UTPS algorithm (refer to the experimental section for a comparison of performance indicators of various classification algorithms).

On the basis of the Bayes theorem, the Naive Bayes algorithm learns a joint probability model for classification prediction via prior and conditional probabilities. Owing to the conditional independence assumption, or in other words, when the category is determined, all features used for classification are conditionally independent, the amount of calculation is significantly reduced. However, such an assumption may not hold in real life, so the classification accuracy of Naive Bayes probably declines.

Naive Bayes is equipped with high training velocity and easiness of generating a classification model, but its classification accuracy is low under scenes related to classification features. Naive

Bayes performs the best among the many classification algorithms to classify feature datasets extracted by the UTPS algorithm. This could be attributed to the limited correlation among the features in the UTPS dataset.

4.5.2 Bagging ensemble

As a fundamental ensemble learning technique, Bagging is a commonly employed method to enhance the performance of classification models when dealing with imbalanced data. The Bagging algorithm has the following steps.

(1) Train a multitude of foundational classifiers. Adopt the bootstrap sampling method and samples from the original dataset to obtain a new dataset; obtain a base classifier by training with a new dataset. Repeat many times to obtain multiple base classifiers.
(2) Integrate multiple base classifiers. Use multiple base classifiers to classify and predict the same samples; adopt a voting mechanism to determine the category most frequently predicted as the final category.

Bagging ensemble of TGH classification model:

As previously discussed, we have chosen to employ random forest for classifying the feature dataset generated by the TGH algorithm. Random forest operates by constructing multiple decision trees, which aligns with the Bagging concept. Consequently, in this study, we do not employ Bagging for ensembling the TGH classification model any further.

Bagging ensemble of UTPS classification model:

The stability of a classifier plays a crucial role in affecting the effectiveness of the Bagging algorithm. Classifier instability implies that perturbations in the dataset can lead to significant fluctuations in classification outcomes. When the base classifier within an ensemble is unstable, Bagging can substantially improve performance.[13] Conversely, the impact is limited if the base classifier is already stable. Fortunately, the Naive Bayes algorithm has been demonstrated to exhibit stability.[25] Therefore, it becomes necessary to induce instability in Naive Bayes to enable Bagging with this base classifier. In this study, we leverage a Bagging Naive Bayes classification approach from the existing literature;[25] it creates diverse training subsets to introduce instability into Naive Bayes and build an ensemble base classifier.

When forming training subsets for each base classifier, the process involves a random selection between two categories, followed by a random sampling of 30% of the samples within the chosen category. Simultaneously, 70% of the samples are drawn randomly from the other category to create a distinct training subset. Consequently, these training subsets exhibit substantial diversity and differ in distribution from the original dataset.

4.5.3 Comprehensive decision-making

We combine two classification models to further improve the model's accuracy and determine the final prediction result via a voting mechanism. The objective is to use this combination to categorize mobile device users associated with the same MAC address. In accordance with the definition, this constitutes a form of ensemble learning in a broader context.

In this study, a label of 1 denotes a person of interest, whereas a label of 0 signifies the general public. When two classification models yield concordant labels, the final prediction adopts their consensus. Conversely, if there is disagreement between the models, the final outcome defaults to 0, indicating the general public. One reason is that two classification models incorrectly predict much of the general public as persons of interest, while such people account for only a small proportion in real life.

Prediction accuracy will be significantly enhanced when the TGH and UTPS are combined in accordance with the above rules. For more details, see the experimental process.

5. Experiment

5.1 Data processing

Concerning MACs collected in a city, the general public accounts for the most significant proportion, and their behaviors differ. Naturally, it is unrealistic to perform statistical analysis on data of the general public. In addition, the number of individuals of the general public is markedly different from that of persons of interest, so the former is undersampled. In this study, a random undersampling technique was employed to select 10653 MAC addresses from a pool of 8 million MAC addresses belonging to regular residents. MAC information of a total of 803 suspects was obtained from relevant departments to constitute an experimental dataset.

In terms of research on space-time trajectories, Gong[26] pointed out that people's activities were mainly concentrated in seven days as one cycle. Consequently, in this study, we conducted classification research on data of one week in August 2017 at the beginning of exploration.

In this study, we explored MAC addresses with seven-day data of the week, including 7,588 general public and 593 persons of interest, as shown in Table 3.

As shown in Fig. 2, the activities of different types of person will show different characteristics at different time periods, especially in the middle of the night, and the activity of persons of interest is much higher than that of the general public.

The trajectory data of different categories show the characteristics of long tail distribution, as shown in Fig. 3.

5.2 Experimental environment

The platform is a Dell server 64-bit system (16 core CPU, each with 2.6 GHz, four GPUs GTX 3090, 32 Gb of main memory). The algorithms and models described herein were implemented by Python 3.7.

Table 3
Initial data preparation.

	Normal residents	Suspects
Total number of collected records	7293153	625421
Total number of experimenters	10653	684
Number of people with data for 7 days	7588	593

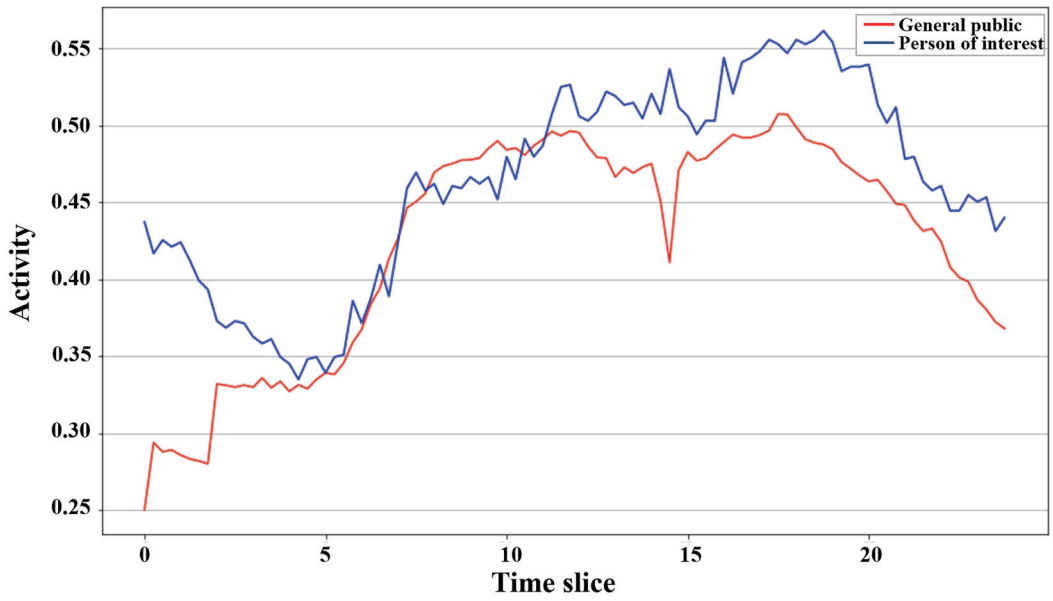

Fig. 2. (Color online) Activities of different types of person at different time periods.

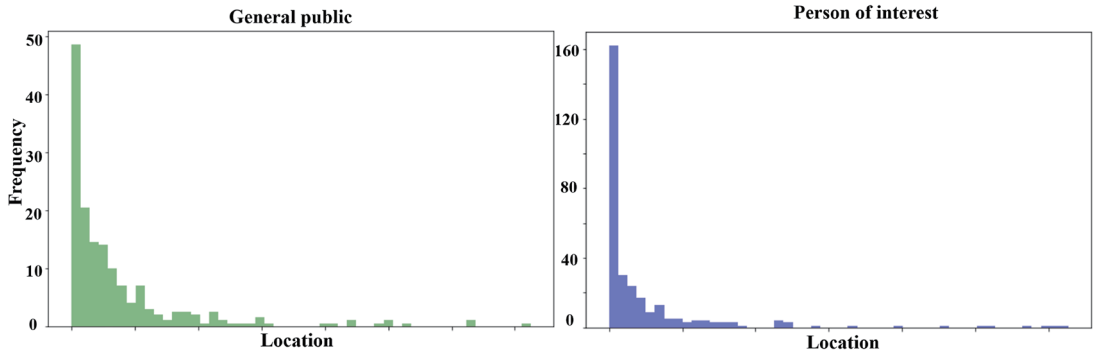

Fig. 3. (Color online) Regional activity distributions of different types of person.

5.3 Feature dimensionality reduction

Table 4 shows the original dataset and dataset dimensions generated by two algorithms.

The TGH has 399 dimensions, including 96 time dimensions and 303 space dimensions. Specifically, the former is formed by the division of 24 h a day, with 15 min as a time slice. The granularity of 15 min is determined through experiments.

Table 5 shows the experimental results obtained under the granularities of 1 h, 30 min, 15 min, and 5 min. The data originate from records collected within one week of MAC addresses of 593 general public and 593 persons of interest selected by the One-Sided Selection undersampling algorithm. The algorithm with default parameters adopts random forest to obtain performance indicators through 10-fold cross-validation.

It is most suitable to select 15 min as the granularity of time slices.

Table 4
Dataset dimensions.

Original dataset	TGH dataset	UTPS dataset
>86400	399	15

Table 5
Results obtained with various time slice division granularities of TGH algorithm.

Granularity	Dataset dimension	*ACC* (%)	*Precision* (%)	*Recall* (%)	*F*1 (%)
1 h	414	81	82	81	81
30 min	438	83	83	82	83
15 min	486	**84**	**85**	**83**	**84**
5 min	678	82	84	83	83

5.4 Classification algorithm selection

In subsequent tests, an oversampling method was found to cause overfitting. Therefore, it was abandoned, and only MAC address samples of the general public were undersampled. The training set obtained finally consisted of MAC addresses of 593 general public and 593 persons of interest.

5.4.1 TGH classification model algorithm selection

As shown in Table 6, random forest used to implement the TGH classification model has the best performance.

5.4.2 UTPS classification model algorithm selection

Ultimately, we opt for the Naive Bayes algorithm as the classification method for the UTPS classification model, because this method can achieve the optimal performance, as shown in Table 7. This choice supersedes more commonly employed classification algorithms, which could be attributed to the limited correlation observed among the features in the UTPS dataset.

Table 6
Performance indicators of common classification algorithms for TGH classification model.

Classification algorithm	*ACC* (%)	*Precision* (%)	*Recall* (%)	*F*1 (%)
Random forest	**84**	83	**86**	**84**
Naive Bayes	75	74	79	77
Logistic	82	83	80	82
SVM	82	**84**	80	82
J48	75	78	71	75
RandomTree	78	78	77	78

Table 7
Performance indicators of common classification algorithms for UTPS classification model.

Classification algorithm	*ACC* (%)	*Precision* (%)	*Recall* (%)	*F*1 (%)
Random forest	79	85	87	86
Naive Bayes	**80**	84	**89**	**87**
Logistic	77	83	87	85
SVM	78	**85**	84	85
J48	77	81	88	85
RandomTree	76	84	83	83

5.5 Bagging ensemble

We leverage a Bagging Naive Bayes classification algorithm described in a prior publication.[25] This algorithm generates diverse training subsets, trains T base classifiers, and ultimately combines these T base classifiers using a voting mechanism.

To determine the appropriate T value, we test on the training set, with results listed in Fig. 4. $T = 1$ means that the Bagging ensemble is not used; T values of other tests are odd numbers, facilitating voting.

Figure 4 demonstrates that increasing the number of base classifiers does not necessarily yield better results. In the case of the UTPS training set, an optimal choice appears to be the use of seven base classifiers.

5.6 Comprehensive decision-making

Tables 8 and 9 show results for the TGH classification model and the UTPS classification model, as well as prediction results and performance indicators after comprehensive decisions.

According to the table, combining the TGH and UTPS classification models for comprehensive decision-making helps further boost prediction accuracy and *F*1.

5.7 Practical significance

Lift in the data mining field is introduced to quantify the changes in ability caused by using or not using the model. *Lift* is calculated as

$$Lift = \frac{TP / (TP + FP)}{(TP + FN) / (TP + FN + FP + TN)}. \tag{8}$$

For the TGH classification model presented in Table 8, the denominator in Eq. (8) corresponds to 0.026. This value signifies that, without employing the model, if an individual were randomly selected from a pool of 7588 general public and 593 persons of interest, the likelihood of being a person of interest would be 2.6%. In contrast, the numerator stands at 0.06, indicating that when utilizing the TGH classification model, if a person were randomly chosen from those predicted

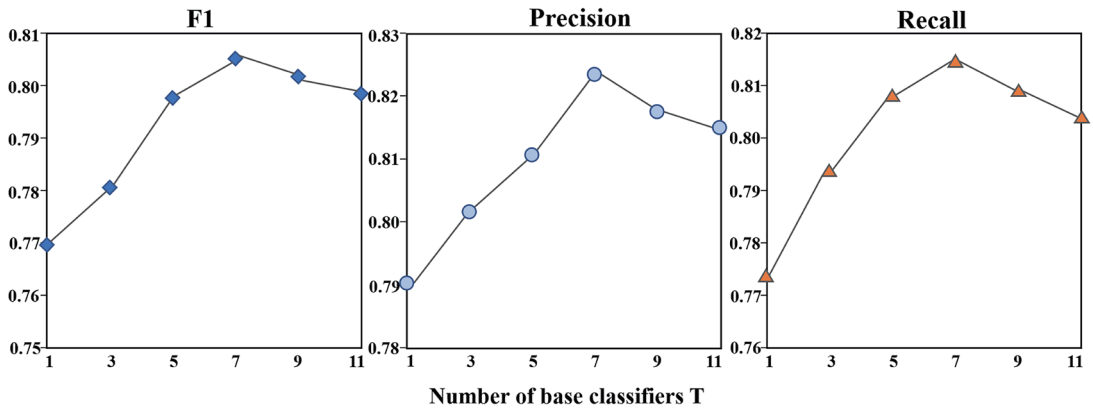

Fig. 4. (Color online) Selection of number of base classifiers for Bagging ensemble of UTPS classification model.

Table 8
Prediction results of classification model on dataset.

Classification algorithm	Predicted people of interest	Actual people of interest	Predicted general public	Actual general public
TGH	2991	179 (6.0%)	5190	5154 (99.3%)
UTPS	2783	175 (6.3%)	5398	5360 (99.3%)
Comprehensive decision	1867	164 (8.8%)	6314	6263 (99.2%)

Table 9
Performance metrics for the dataset in the classification model.

Classification algorithm	*ACC* (%)	*Precision* (%)	*Recall* (%)	*F*1 (%)
TGH	66	7	84	12
UTPS	68	6	83	12
Comprehensive decision	80	9	78	16

by the model as persons of interest, the probability of actually being a person of interest would be 6.0%. The resulting *lift* value of 2.3 demonstrates that, with the TGH classification model in use, the likelihood of selecting a person of interest from those predicted is 2.3 times higher than in the original dataset. This *lift* value further increases to 3.4 when combining the TGH and UTPS classification models.

Owing to the limited number of samples for persons of interest, the classification model's accuracy currently falls below 80%. However, the *lift* value signifies that the probability of an actual person of interest among people predicted to be a person of interest has increased several times. This is conducive to better analyzing and classifying different groups of people.

To recap, we introduce the TGH and UTPS algorithms for feature extraction. Coupled with the ongoing refinement of our classification model, we aspire to provide guidance for future research endeavors in this domain. The findings from this phase have already found practical applications within real-world systems.

6. Conclusions

Using the vast spatiotemporal data from city sensors, we explore the classification prediction of a person of interest using spatiotemporal data in traditional machine learning methods. We shall provide ideas for similar research. Unfortunately, owing to uncertainties in machine learning, methods should be improved and expanded, including the combination of classification algorithms and the processing of unbalanced datasets. In this paper, we limit the number of persons of interest, which may hinder the improvement of the classification model performance. Therefore, we shall continue investigations in this field and constantly improve classification models after acquiring more samples of persons of interest.

Author Contributions

Conceptualization, L. C. and C. J.; methodology, L. C.; software, L. C.; validation, L. C., H. W., and J. Z.; formal analysis, H. W. and J. Z.; investigation, L. C.; resources, C. J.; data curation, H. W. and J. Z.; writing—original draft preparation, L. C.; writing—review and editing, L. C., C. J., and H. W.; visualization, J. Z.; supervision, C. J. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This research was funded by the Natural Science Foundation of Zhejiang Province of China (Grant No. LY21F020003), the Zhejiang Science and Technology Plan Project (Grant No. 2021C02060), and the Scientific Research Foundation of Hangzhou City University (Grant No. X-202206).

References

1 Y. Li and S. Wu: Cyber Secur. Data Governance **33** (2014) 3. http://doi.org/10.19358/j.issn.1674-7720.2014.11.001
2 C. Jin, H. Liang, D. Chen, Z. Lin, and M. Wu: Proc. Advances in Knowledge Discovery and Data Mining: 23rd Pacific-Asia Conf. (PAKDD 2019) 477–488. https://doi.org/10.1007/978-3-030-16148-4_37
3 C. Jin, D. Chen, Z. Lin, and M. Wu: GeoInformatica **25** (2021) 799. https://doi.org/10.1007/s10707-021-00448-9
4 C. Jin, T. Tao, T. Ruan, L. Xu, X. Luo, and R. Li: ICICAS (IEEE 2019) 638. https://doi.org/10.1109/ICICAS48597.2019.00139
5 B. Du, C. Liu, W. Zhou, Z. Hou, and H. Xiong: Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining. (2016) 87–96. http://doi.org/10.1145/2939672.2939687
6 C. Song, Z. Qu, N. Blumm, and A. L. Barabási: Science **327** (2010) 1018. http://doi.org/10.1126/science.1177170
7 C. Song, T. Koren, P. Wang, and A. L. Barabási: Nat. Phys. **6** (2010) 818. https://doi.org/10.1038/nphys1760
8 Y. A. De Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel: Sci. Rep. **3** (2013) 1. https://doi.org/10.1038/srep01376.
9 J. L. Toole, C. Herrera-Yaqüe, C. M. Schneider, and M. C. González: J. Royal Soc. Interface **12** (2015) 20141128. https://doi.org/10.1098/rsif.2014.1128
10 J. H. Friedman: Data Min. Knowl. Discovery **1** (2002) 55. https://doi.org/10.1023/a:1009778005914
11 Q. Yun: Research on Application of Classification Algorithms for Imbalances Data. Ph.D. thesis, Changchun: Jilin University (2014).
12 L. Breiman: Mach. Learn. **24** (1996) 123. https://doi.org/10.1007/BF00058655

13 M. Galar, A. Fernandez, B. E. Arrenechea, H. Bustince, and F. Herrera: IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **42** (2012) 463. https://doi.org/10.1109/tsmcc.2011.2161285
14 S. Dodge, R. Weibel, and P. Laube: SIGSPATIAL Special **1** (2009) 11. https://doi.org/10.1145/1645424.1645427
15 Y. Yanagisawa, J. Akahani, and T. Satoh: Proc. Mobile Data Management: 4th Int. Conf. MDM 2003 (2002) 63–77. https://doi.org/10.1007/3-540-36389-0_5
16 M. Vlachos, G. Kollios, and D. Gunopulos: Proc. 8th Int. Conf. on Data Engineering (IEEE 2002) 673–684. http://doi.org/10.1109/ICDE.2002.994784.
17 L. Chen, M. T. Özsu, and V. Oria: Proc. 2005 ACM SIGMOD Int. Conf. Management of Data. (2005) 491–502. https://doi.org/10.1145/1066157.1066213.
18 C. Jin, T. Tao, X. Luo, Z. Liu, and M. Wu: IEEE Access **8** (2020) 58763. https://doi.org/10.1109/ACCESS.2020.2982823
19 Y. Sakurai, M. Yoshikawa, and C. Faloutsos: Proc. 24th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems (2005) 326–337. https://doi.org/10.1145/1065167.1065210.
20 E. Frentzos, K. Gratsias, and Y. Theodoridis: Proc. 2007 IEEE 23rd Int. Conf. Data Engineering (IEEE 2007) 816–825. http://doi.org/10.1109/ICDE.2007.367927.
21 J. Cuenca-Jara, F. Terroso-Saenz, M. Valdés-Vela, and A. F. Skarmeta: Appl. Soft Comput. **86** (2020) 105916. https://doi.org/10.1016/j.asoc.2019.105916.
22 Y. Li, H. Liu, X. Zheng, Y. Han, and L. Li: IEEE Access **7** (2019) 29679. http://doi.org/10.1109/ACCESS.2019.2902310.
23 J. Zhao, B. Chen, L. Yang, and Y. Yao: Sci. Technol. Eng. **13** (2013) 80.
24 Y. Zhou, Y. Li, Y. Huang, and E. Geng: J. Geo-Inf. Sci. **19** (2017) 1238.
25 J. Suqin, S. Hongbo, and W. Jie: Comput. Eng. **38** (2012) 203.
26 W. Gong: MA thesis, Shanghai Jiao Tong University (2014). https://kns.cnki.net/KCMS/detail/detail.aspx?dbname=CMFD201501&filename=1015028127.nh.
27 M. Kearns and L. Valiant: J. ACM (IJCM) **41** (1994) 67. https://doi.org/10.1145/174644.174647.

About the Authors

Lifeng Chen works in the Information and Technology Center (Supercomputing Center) of Hangzhou City University and is mainly in charge of the construction, operation, and maintenance of major digital applications of the school, such as the school's public data collaboration platform, the design and development of campus mobile and PC applications, and the online teaching platform, with rich experience in project construction and management. He is currently serving as the digital commissioner of Hangzhou City, connecting the school with the digital reform task of Hangzhou City. (chenlf@hzcu.edu.cn)

Canghong Jin is an associate professor of computer science at Hangzhou City University. His research focuses on mining and modeling large social and information networks, spatiotemporal series mining, and the big data platform. The problems he investigates are motivated by large-scale transit records, the web, and online media. (jinch@hzcu.edu.cn)

Hao Wu received his B.S. degree from Hangzhou City University, China, in 2021. He is currently pursuing his M.S. degree in computer and information systems at Macau University of Science and Technology, China. His research interests include graph neural networks, computer vision, and artificial intelligence, and he is currently working on object tracking. (2220024311@student.must.edu.mo)

Jiafeng Zhao received his B.E. degree from Hangzhou Normal University, China, in 2021. He is currently pursuing his M.E. degree at Zhejiang University of Technology, China. His research interests include graph neural networks and artificial intelligence. (2112112172@zjut.edu.cn)

Jianghong Wu is the deputy director of Zhejiang Key Laboratory of Social Security Big Data. His research focuses on the application of big data and AI in the prevention and control of social risks. (13867180080@139.com)