3-2020

S2N2: An interpretive semantic structure attention neural network for trajectory classification

Canghong JIN

Ting TAO

Xianzhe LUO

Zemin LIU
Singapore Management University, zmliu@smu.edu.sg

Minghui WU

Citation

JIN, Canghong; TAO, Ting; LUO, Xianzhe; LIU, Zemin; and WU, Minghui. S2N2: An interpretive semantic structure attention neural network for trajectory classification. (2020). *IEEE Access*. 8, 58763-58773.
Available at: https://ink.library.smu.edu.sg/sis_research/5143

IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 28, 2020, accepted March 11, 2020, date of publication March 23, 2020, date of current version April 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982823

S2N2: An Interpretive Semantic Structure Attention Neural Network for Trajectory Classification

CANGHONG JIN[1], TING TAO[1], XIANZHE LUO[1], ZEMIN LIU[2], AND MINGHUI WU[1]

[1]Zhejiang University City College, Hangzhou 310012, China
[2]School of Information Systems, Singapore Management University, Singapore 188065

Corresponding author: Minghui Wu (mhwu@zucc.edu.cn)

This work was supported by the Research Support Project of Education of Zhejiang Province under Grant Y20194137.

ABSTRACT We have witnessed a rapid growth over past decades in sensor data mining (SDM), which aims at extracting valuable information automatically from large repositories of moving activity data. One of the significant SDM tasks is identifying humans through their transit modes using a variety of user-tracking systems. However, to the best of our knowledge, distinguishing traces of users and understanding their behaviors are difficult tasks in most real-life cases for the following reasons: 1) activity data containing both temporal and spatial contexts are of high order and sparse; 2) living patterns are not as regular as expected, and the route choice uncertainties due to their *vagueness* and *randomness* ; 3) owing to the complexity and sparseness of urban travel methods, although some deep learning-based models can produce relatively good classification results, they can still be improved by combining external information. To address these challenges, we propose a novel scenario-based deep learning method which is based on the assumption that people visit places with explicit purposes (e.g., to go to work or visit a park). We first represent semantic patterns from daily life and create various scenarios and utilize an attention neural network to embed points of trajectories by considering both semantic and geographical information. Then, we construct a Semantic Structure Neural Network (S2N2) framework to perform the end–end classification. Our S2N2 model is applied to an interesting yet challenging topic: distinguishing suspect transit behavior on a real-life data set collected by mobile devices. Although the problem is not entirely solved, the extensive evaluation presented here demonstrates that our model outperforms conventional classification methods, anomaly detection methods, and state-of-the-art sequential deep learning models, especially when trajectory semantic vectors are incorporated. We also provide statistical analysis and intuitive explanations to help interpret the characteristics of user mobility.

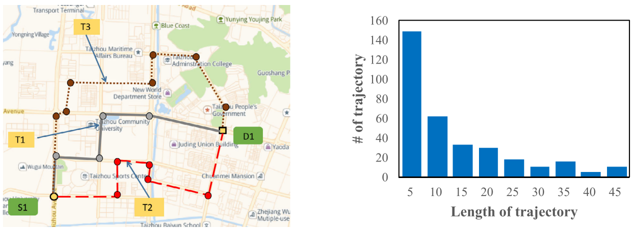
INDEX TERMS User classification, interpretive trajectory structure, human behavior understanding.

I. INTRODUCTION

Advances in location-acquisition techniques and the prevalence of location-based services have generated a massive amount of spatial trajectory data, which represent the mobility of a diversity of moving objects, such as people, vehicles, and animals. Due to the recent cloud environment's achievements in resource allocation algorithms and minimizing workflow execution costs, processing large-scale trajectory data has become more efficient [1], [2]. Estimating users' contexts from their movement trajectories obtained from devices such as mobile phones with GPS is crucial for location-based services, and understanding knowledge obtained from trajectory data allows us to draw an overall picture of human activities and extract regular life patterns from it.

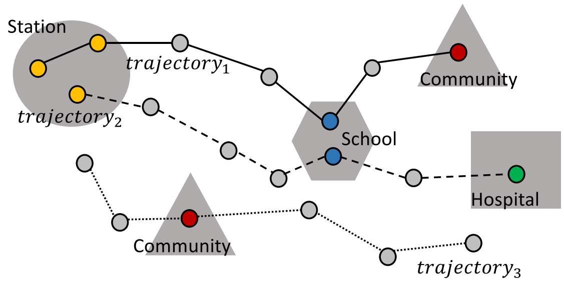
Trajectory pattern mining through transit modes is of great importance for a broad spectrum of tracking systems, e.g., transportation mode mining, urban planning, and recommendation systems. With the development of location recognition technology, the end-to-end recognition classification model proposed in [3] has obtained a fairly high accuracy rate in this field. This is important for enriching the background significance of the trajectory. In particular, object identification

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



(a) paths between two points *(b) different lengths of paths*

FIGURE 1. Uncertainty of urban trajectories.



ID	# of points	trajectory semantic	similarity by *Jaccard*
1	7	station – school – community	sim(1,2) = 0.5, sim(1,3) = 0.3
2	7	station – school – hospital	sim(2,1) = 0.5, sim(2,3) = 0
3	6	community	sim(3,1) = 0.3, sim(3,2) = 0

FIGURE 2. The motivation of S2N2. There are three trajectories and four different semantic scenarios.

has become increasingly popular with the development of computing technology and the availability of trajectory data containing sufficient semantic and geographic information. For example, in Beijing the police turn to Google to find crime suspects by seeking data from mobile phones in target areas, and the identification of suspected pickpockets has been realized by mining public transport records [4]. However, the data of automated fare collection systems (AFC) are following fixed routes, in our situation, people travel in the city, and their movement behaviours are more complicated. Therefore, it is very important to propose a new monitoring and a tracking framework to detect all kinds of users.

Unfortunately, in many cases, it is not a trivial task to identify users from their travel records. On the one hand, it is because collected trajectories are sparse and diverse, on the other hand, there are uncertainties in behavior choices in an urban environment. The uncertainties that transportation researchers frequently encounter and are uncomfortable dealing with can be divided into two different types [5]. One is *randomness* due to the nondeterministic nature of choice behavior problems and random utility models based on probability distributions have been employed to deal with this randomness. The second type of uncertainty is *vagueness* due to a lack of familiarity with road networks and the linguistic information associated with network attributes [6]. Song *et al.* [7] has studied millions of users and has pointed out that the movement patterns of users can easily appear random and unpredictable.

In this paper, we will use a case study to illustrate the characteristics of urban trajectories in Fig. 1. We collect urban movement data of users from their mobile phones, including spatial and temporal information. Moreover, each place has its unique category by its installed place, such as hotel, roadside, park, or railway station. For example, we choose two frequently visited locations named S_1 and D_1, and note that there are several different trajectories between two points, such as T_1, T_2, and T_3. We further illustrate the randomness and vagueness of route choice by statistical evaluations, we calculate the lengths of all trajectories between S_1 and D_1 and show that from an observation point of view, there are many various length paths through these two points and they are quite different. To further illustrate the uncertainty of trajectories, we select and calculate some major travel features of trajectories such as velocity, scope of activity, curve rate, speed, and HCR in Fig. 3, from which we can see the diverse distributions of all the features.

Generally, trajectory feature-based classification applications, extracting effective transit features was a fundamental task. The purpose and behavior of a user traveling through the city are considered as features of trajectories (i.e., trajectory length, duration, head change rate (HCR), and velocity change rate (VCR)) [8]. Based on previous studies, feature-based trajectory classification gives a better performance in the case of high-precision continuous data, such as GPS signals. Instead of GPS, the WiFi sensor data in our study are discrete,*vagueness* and *randomness* mentioned in Fig. 3, which prevents the acquisition of effective trajectory features for feature-based models.

Some recent deep-learning-based methods embed every point of a trajectory into a unique vector, present each trajectory as a sequence of points and could improve the accuracy of many trajectory tasks [9]–[11]. In some specific cases, the accuracy of RNN classification based on TrajectoryNet method [12] can even reach above 95%. However, if we consider a trajectory to consist of a series of points, all the visited points are treated the same, and the semantics of the trajectory is a negligible part. Thus, when understanding each trajectory, it is necessary to capture these path-concept and semantic-concept associations. Our experiment in Section VI also demonstrates that neither traditional machine learning methods nor recurrent neural network methods are able to achieve excellent performance for whole trajectories.

To address the challenges mentioned above, in this paper, our solution is to select and integrate semantic scenarios with trajectories during the learning process. For example, in Fig. 2, there are three trajectories and four location categories. Trajectory 1 has the semantic path ''station–school–community'', whereas trajectory 3 has only semantic ''community''. Under the path view, the points that three trajectories pass are discrete and totally different, thus the trajectories have the sparse, randomness and vagueness problems. But if under the semantic view, couple $< trajectory1, trajectory2 >$ has closer semantics

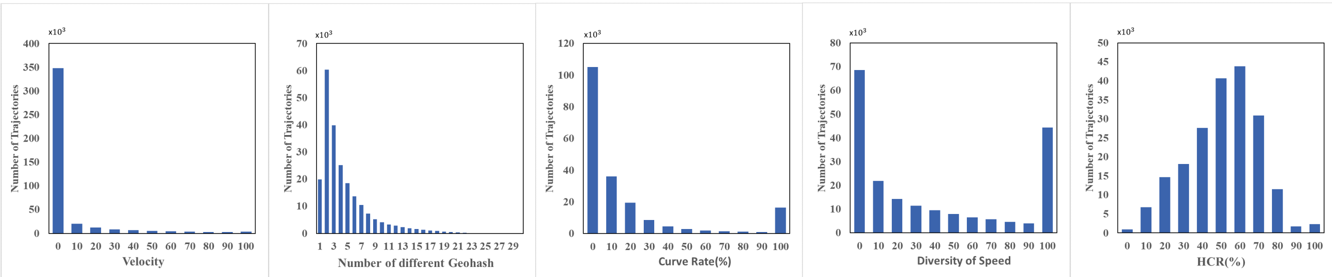


FIGURE 3. Number distributions of motion features in the city.

than couple $< trajectory1, trajectory3 >$, which could build a novel representation for classifying and understanding mobility.

Based on above motivation, we propose a Semantic Structure Neural Network(S2N2) model to select some target extensional scenarios, pre-trained points with semantic information, and utilizes trajectory vectors for end-to-end classification, which could significantly improve the performance of models.

Our contributions are summarized as follows:

- We propose an end-to-end framework containing well-selected scenarios and embedding with the aim of classifying individuals by their long-range and sparse trajectories. Our framework is generally applicable to all classification problems and does not depend on human-selected features.
- We leverage attention-based network to represent the points of trajectories by using two tasks: masked point model and segment similarity prediction model. After that, we use trajectory structure information as joint features for learning. To the best of our knowledge, our framework is the first to identify users by deep semantic trajectories.
- We evaluate the precision, recall, and F-score on a real-life mobility dataset. Compared with the results of classical classification methods, anomaly detection methods, and state-of-the-art deep learning models, such as long short-term memory (LSTM) in sequential models, the evaluation results for our model demonstrate that it outperforms all the baselines.
- We give the explanations to classification models, including propose LIME and SHAP values as a unified measure of feature importance and demonstrating which scenarios are more effective to describe human mobility. Our explanation is simple but easy to aid decision makers to understand trajectories in critical semantic domains.

The remainder of this paper is organized as follows. Section II provides a brief review of related work. Section III presents the problem definition and notation. Section IV gives a detailed description of the S2N2 framework, and this is followed in Section V by a description of the experiment. In Section VI, we discuss how our classification approach can be used to understand user travel behavior. Finally, we conclude the paper in Section VII.

II. RELATED WORK

Spatiotemporal pattern mining has emerged as an active research field, with examples including urban traffic network analysis, automatic intersection recognition, and movement behavior mining. In this section, we provide a brief review of related work, including three categories: movement pattern mining, trajectory classification and behavior understanding.

A. MOVEMENT PATTERN MINING

The enormous amount of spatiotemporal data that are available can be used to mine movement patterns. Gong *et al.* [13] proposed a methodology to detect five travel models (walk, car, bus, subway and commuter rail) from the amount of data generated by GPS in New York. In [14], Pinelli and co-workers proposed an extension of the sequential pattern mining paradigm to analyze the trajectories of moving objects. The REMO (relative motion) method [15] is based on a traditional cartographic approach of comparing snapshots and is a comparison method based on the use of motion parameters to reveal movement patterns. In [16], a complete and computationally tractable model was presented for estimating and predicting trajectories based on sparse sampling and anonymous GPS landmarks called GPS snippets. In [17], spatiotemporal patterns were identified from GPS traces of taxis for night bus route planning. In [18], an attempt was made to reflect the common routing preferences of previous passengers by finding the most frequent path in a certain period. The approach described in [19] discovers and explains movement patterns of a set of moving objects (e.g., track management, bird migration, or spread of disease). These previous works have given us inspiration for representation of a trajectory, and the traditional machine learning approaches proposed therein have been used to provide baselines for comparison in Section V.

B. TRAJECTORY CLASSIFICATION

A number of techniques for detecting user behavior have also been proposed. For example, the approach presented in [4] extracts user features from subway transit records and explores abnormal travel behavior to discover pickpocket suspects. In the context of location-based anomaly detection, a framework that learns the context of different functional regions in a city was presented in [20], and this provides the basis of our feature extraction approach. Traditional trajectory-based similarity calculations use the longest common substring to calculate the similarity of user history trajectories [21], [22]. Abul *et al.* [23] proposed a W4M (wait for me) method, which uses the edit distance to measure the similarity of different paths. Considering the mobility similarity between user group, Zhang *et al.* [24] proposed the GMove modeling method to share significant movement regularity. In recent research, some deep learning methods have been applied to encode the trajectory. ST-ResNet [9] is designed to forecast the flow of a crowd. The DeepMove [10] model predicts human mobility with a recurrent attentional network, and HST-LSTM [11] captures location prediction by spatiotemporal LSTM. Our approach shares some aspects with the abovementioned embedding techniques, but, unlike classical classification models, our model describes users' behaviors on the basis of their movement scenarios and classifies users by a convolutional neural network without selecting effective features.

C. SEMANTIC TRAJECTORY AND BEHAVIOR UNDERSTANDING

Several data models have been proposed for efficiently querying raw trajectory data [25], [26], but only a few approaches are able to consider a trajectory together with its background geographic information and deep meaning. In [27], it was pointed out that mining trajectory data should not focus on trajectories alone, but should also utilize the rich contexts from the background to provide a semantic understanding of these trajectories. In [28], a semantic model was proposed for trajectories as well as for the relationships of trajectories with background geographic information. However, this model is applicable only for special areas and has serious limitations. Another problem is that none of these models are easy to interpret. Therefore, in this paper, we use GEOHASH to demarcate a range and divide specific trajectory points into regional blocks with specific meanings. The resulting trajectory uses changes in regional block type as its semantics and fuses these with the semantics of the original trajectory, thereby combining trajectory and trajectory background information and making them easier to interpret during analysis.

Human behavioral characteristics exhibit regularity, persistence, and other characteristics. The behavior in which a person often engages can reflect their preferences, personality characteristics, and even temperament to a certain extent. Reference [29] proposed an algorithm for identifying human behavior through image features, but it is difficult to obtain good behavior understanding under our trajectory dataset. We obtain a large amount of user behavior information [30], after user trajectory feature extraction and preprocessing. This includes location, duration, and other information. After analysis, we can mine a large amount of similar user behavior characteristic information from massive data, and classify users through these similar trajectories, in order to achieve the purpose of distinguishing different types of followed groups.

TABLE 1. Some important notation.

Notation	Description
u	A user u, who is a resident or a suspect
Ty	The type of a trajectory(resident or suspect)
$O_i^u(g,k)$	Moving data of a certain people u and consisting of location g and location category k
$\mathcal{P}_i^u(O)$	A trajectory $\mathcal{P}_i$ belonging to a certain people category u and consisting of a series of O_i
$\mathcal{S}^u(\mathcal{P})$	A semantic block of trajectory P_i belonging to a certain people category u
SIM	The similarity between two trajectories
$\mathcal{T}_i$	Location types of key point
d	Dimensionality of the embedding

III. PRELIMINARIES

We introduce the definitions of several basic concepts and provide a formal definition of the scenario-based user identification problem. The notation that we use is summarized in Table 1.

Definition 1: A **trajectory** $\mathcal{P}_i^u = \{O_1, O_2, \ldots, O_n\}$ consists of a series of points belonging to a certain people u, and each point can be represented as $O_i^u(g, k)$, abbreviated as O_i, where g is position information (latitude and longitude) and k is the type of this point, which is used to construct semantics.

For instance, in Fig. 2, $\mathcal{P}_1^u = \{O_1, O_2, O_3, O_4, O_5, O_6, O_7\}$ represents the first trajectory where O_i means the $i-th$ point in this trajectory.

Definition 2: A **semantic segment** $\mathcal{S}^u(\mathcal{P}_i)$ represents a semantic block of trajectory P, which is named P_i and belongs to a certain people u. It is a list, the length of which is the same as the length of the trajectory. In Fig.2, $\mathcal{S}^u(\mathcal{P}_1) = \langle$*gas station, gas station, street, street, school, street, home community*$\rangle$ represents the semantic block of the trajectory $\mathcal{P}_1^u$

Combining Definitions 1 and 2, we can accurately describe the trajectory characteristics of a trajectory and its semantic characteristics.

Definition 3: A **type of key point** $\mathcal{T}_i$ represents the location type of a key point and We can obtain some specific motion patterns from it.

Because in the original data more than 60% of the sensors were placed in places that have little information, such as intersections, roadsides, etc., we consider that the remaining points in the trajectory after removing these locations are key points.

Definition 4: The **user classification** problem is defined as the classification of whether a given user u is a suspect or not and find out the Ty of this user by giving trajectories $\langle\mathcal{P}\rangle$ and the semantic block $\langle\mathcal{S}\rangle$ of the trajectory belonging to this user.

In this paper, our data include two different people categories called residents and suspects. We focus on residents

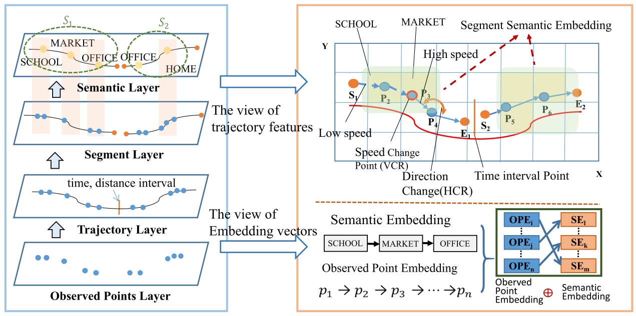


FIGURE 4. Two types of trajectory classification approaches: one is feature based and the other is deep learning based.

who have long-range and high-quality trajectory data. Moreover, in order to generate discriminative features of mobility, we compress the spatial and temporal information of trajectories by using PRESS [31], which proposes a Hybrid Spatial Compression (HSC) algorithm and error Bounded Temporal Compression (BTC) algorithm respectively.

IV. S2N2 FRAMEWORK

In this section, we introduce the major concepts of the S2N2 framework, including 1) scenario selection and latent similar trajectories cluster model; 2) attention based trajectory embed and fusion model; 3) the semantic combined end-to-end classification model.

A. DETAILS OF S2N2 FRAMEWORK

1) SEMANTICS OF TRAJECTORIES

To generate an intermediate representation of the moving record that is less noisy and more suitable for the process of understanding motion patterns, we extract the semantics of trajectories from visited location categories, and we find observed motion patterns via some frequent item mining methods, which find interesting subsequences or substructures in a large-scale moving data set.

We map every point in a trajectory to its related location category, and generate a *semantic path*. We counted the number of occurrences of each place in the raw data and then we perform the following two steps to integrate the data as much as possible while retaining the meaning of the place:

Step1:Blur the details of the original place, such as turning the "Sailing Hotel" into "Hotel", making the clustering more universal.

Step2:Abstract similar places into general representative areas, such as "high school" and "primary school" collectively referred to as "school".

After that we can get 15 labels. In these 15 labels, because of most sensors are installed on the roadside and lack clear location semantics we remove these "street" points. So in the end we separate points into other 14 labels: *hotel*, *government*, *Internet cafe*, *labor market*, *home community*, *supermarket*, *gas station*, *shopping mall*, *police station*, *hospital*, *scenic spot*, *cinema*, *school*, or *bank*. Every trajectory can then be represented by one of these semantic categories. For example, in Fig. 2, the semantics of trajectories 1 and 2 are *gas station–school–home community* and *gas station–school–hospital*. We then find motion patterns via a frequent mining approach, such as *a priori*-based approach or a pattern-growth approach. Either of these methods is suitable for our framework. Each segment has a related set of labels. For instance, trajectory 1 has ⟨*gas station, school, home community*⟩.

Finally, several similarity functions for trajectories are given to measure the similarity of different trajectories, including dynamic time warping (DTW), longest common subsequence (LCSS), and modified Hausdorff distance. Since the lengths of segments in our study are relatively small and most points have the same label "street", merely using DTW or LCSS is not suitable for direct measurement. In the end, we decided to use a method called *unite-similarity* which combined DTW and Jaccard to calculate the similarity between trajectories. For example, after processing the original trajectory data ($\mathcal{P}_1^u(O)$ and $\mathcal{S}^u(P_1)$) by deleting the invalid points, we can get two trajectories: $t_1 = \langle p_1, p_2, \ldots, p_n \rangle$, with labels $\langle \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n \rangle$, and $t_2 = \langle q_1, q_2, \ldots, q_m \rangle$, with labels $\langle \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m \rangle$. Then we use Jaccard distance to calculate the similarity of semantic features between these two trajectories (in algorithm 1:line 2-6), and use DTW to calculate the similarity of trajectory features between them (in algorithm 1:line 7-13) and then form a two-dimensional vector. Then the Euclidean distance from this vector to the point (0,0) is used to measure the *unite-similarity* between the two trajectories (in algorithm 1:line 14). So for algorithm 1, its time complexity consists of two parts. One of them is the time complexity for calculating Jaccard similarity is about $O(n)$, and the other is the time complexity for calculating DTW is about $O(n^2)$. So the total time complexity of algorithm 1 is $O(n^2)$ (n represents the average length of the trajectories).

2) TrajBERT: ATTENTION PRETRAINING MODEL

In natural language tasks such as question answering (QA) and natural language inference (NLI), the pretraining models such as BERT (bidirectional encoder representations from transformers) achieve the greatest success. Here, inspired by BERT, we propose a novel method named TrajBERT to train the vectors of each point in a trajectory with semantic information. In our model there are two subtasks: the masked point model and segment similarity prediction model.

Masked Point Model: Although the masked language model (MLM) is strictly more powerful than the single model and the bidirectional directional model, the randomly select strategy in original mask procedure ignores the semantics and weight of points in the trajectory and would reduce the performance of classification due to the randomness and vagueness situations.

Furthermore, since the one-shot representations for points and semantic labels are too sparse to train, we utilize an embedding operation to convert the initialized vectors to low-dimensional vectors with dense values. We take the label c_i as example, and in this case the converted vector e_i is

Algorithm 1 Similarity in Trajectory

Input: Two different trajectories and their semantic block, $\mathcal{P}_1^u(O)$, $\mathcal{P}_2^{u'}(O)$, $\mathcal{S}^u(\mathcal{P}_1)$ and $\mathcal{S}^{u'}(\mathcal{P}_2)$

Output: A real number *SIM* representing the similarity of these two trajectories

Initialize: $Mc \leftarrow \infty$; n = length($\mathcal{P}_1$); m = length($\mathcal{P}_2$);
$A \leftarrow unique(\mathcal{S}^u(\mathcal{P}_1))$ //*unique* means removing duplicate data to ensure element uniqueness
$B \leftarrow unique(\mathcal{S}^{u'}(\mathcal{P}_2))$
$k1 \leftarrow size(A \cap B)$
$k2 \leftarrow size(A \cup B)$
$Y \leftarrow 1 - k1/k2$
for each point $O_i \in \langle \mathcal{P}_1^u(O) \rangle$ **do**
  for each point $O_j \in \langle \mathcal{P}_2^{u'}(O) \rangle$ **do**
    $dismat[i][j]$ = getEuclideanDistance($O_i(g)$, $O_j(g)$)
    $Mc[i][j] = min(Mc[i-1][j], Mc[i][j-1], Mc[i-1][j-1])$
  end for
end for
$X \leftarrow Mc[n][m]$
$SIM \leftarrow getEuclideanDistance((X, Y), (0, 0))$
return SIM

represented as

$$e_i = c_i W_u, \tag{1}$$

where $W_u \in \mathcal{R}^{L \times d_2}$ are the parameters of the embedding layer.

Then, we select masked tokens with their related category labels. We refer to this procedure as "semantic masked point." In all of our experiments, we set a parameter δ to decide whether a point should be masked. For each segment in a trajectory, we replace the i-th point with the [MASK] token if the generated random value is larger than δ. Moreover, we select entities from points and choose parts of entities at random. To reflect the correlation between location labels and duplicated points in a trajectory, we mask these points by [MASK] at the same time and predict these masked points by corresponding hidden vectors.

In our training procedure, the strategy of point replacement is similar as in BERT. We replace the chosen point by the following rules: 1) the token [MASK] 80% of the time; 2) a random point 10% of the time. Then, the transformer function is used to predict the original point or label with cross entropy loss.

Segment Similarity Prediction: In order to consider the latent semantic during the pre-train procedure, segment similarity prediction task is created to understand the *relationship* of segments.

Compared with next sentence prediction in BERT, the length of a segment is shorter than the text corpus, thus we design a strategy to generate the relationships of different trajectories by their semantic motion patterns. In this work, we use algorithm 1 to calculate the semantic similarity value of two trajectories and note that the larger the SIM value we get, the greater the difference between them. In terms of similarity between a selected trajectory and all other trajectories, we utilize SIM as a basis to extract pairs of segment, in which segments in the same pair have more similar semantics than segments in different pairs.

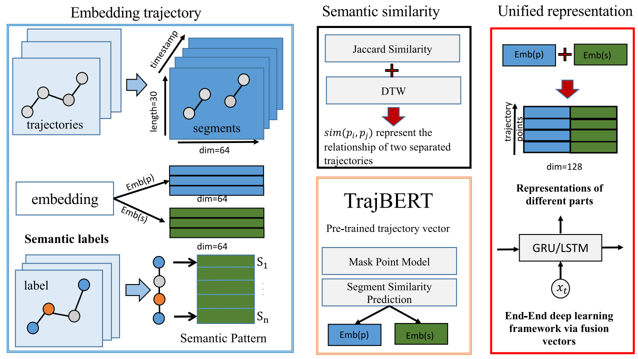


FIGURE 5. Details of S2N2 method.

In our paper, when choosing the folds A and B for each example, 50% of the time B is the actual related trajectory that follows A (labeled as similar latent semantic), and 50% of the time it is an irrelevant trajectory from the data set(labeled as dissimilar latent semantic). Segment similarity prediction task is designed as a binary classification task and is closely related to representation learning objectives used in [32]. By using pretrained embedding, like prior work, only point embeddings are transferred to classification downstream task, where we transfers all parameters to initialize End2End model parameters.

B. SEMANTIC FUSION AND End2End CLASSIFICATION

Motivated by the success of embedding techniques in other areas, we now utilize deep neural network methods to classify trajectories. After preprocessing movement records and type of location by the TrajBERT and W2V methods, We integrate trajectory features with their location semantic features to generate a vector of trajectories and a vector of semantics and both of them are 64-dimensional. We then concatenate these two vectors as a new 128-dimensional vector that provides a better reflection of the characteristics of the trajectories and human behavior than any previous vector.

Here, we show how this input vector X, which obviously includes both trajectory and semantic features, is obtained:

$$X_k = P_k \oplus S_k, \tag{2}$$

where $\oplus$ is the operation that concatenates two vectors into a long vector, P_k is the vector of trajectories that are learned by TrajBERT, and S_k is the vector of semantics that are trained by W2V.

After that, we use the Gated Recurrent Unit (GRU) model to train and predict the data that has been preprocessed. This model is simpler than the standard LSTM model. The effect is similar to LSTM, but the parameters are 1/3 less, and it is not easy to overfit. The biggest advantage of GRU is simplicity (because there are only two gates), and the computational overhead is small, which is more suitable for large-scale data sets. And r_t is the reset gate and Z_t is the update gate [33].

The update gate Z_t is calculated first and its value is between 0 and 1:

$$Z_t = \theta(W_z * [h_{t-1}, x_t]) \tag{3}$$

where h_{t-1} is the previous hidden state and x_t is the current input. Then, a sigmoid function is used to obtain a result of 0 to 1, which determines how much information is retained in the previous hidden state and how much content needs to be remembered. Then we calculate the reset gate:

$$r_t = \theta(W_r * [h_{t-1}, x_t]) \tag{4}$$

It resets h_{t-1}, that is, how much information needs to be forgotten, and then sends it to the *tanh* function with the current input x_t to get the new memory content $\widetilde{h_t}$:

$$\widetilde{h_t} = tanh(W_{\widetilde{h}} * [r_t * h_{t-1}, x_t]) \tag{5}$$

The memory calculation formula for the current time step is:

$$h_t = (1 - Z_t) * h_{t-1} + Z_t * \widetilde{h_t} \tag{6}$$

V. EXPERIMENTS

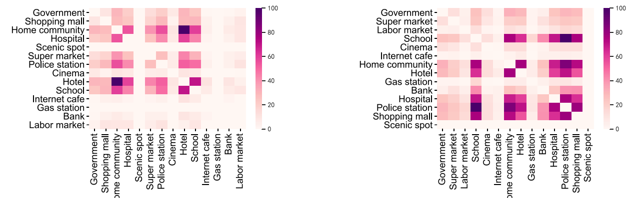
A. DATASET DESCRIPTION

Our real-life dataset was collected by WiFi sensors installed in an eastern city of China and includes MAC address, timestamp and geo-information. Due to the huge amount of moving data and privacy considerations, we only selected a small number of MACs randomly as negative instances. While all the MACs of suspects are selected as positive instances. Since our study focuses on the mobility pattern of trajectories, we needed to do preprocessing to choose actual residents in two steps. First, according to the top-10 best-selling and most popular phones in China, we checked if the MAC belonged to one of the following Android brands, which accounted for over 75% of the market: Huawei, OPPO, Vivo, Xiaomi, Meizu, Gionee, Samsung, Letv, and Lephone. We then selected those residents whose MACs had enough activity track records in total and have at least two weeks of data in a month. As a result, in the dataset, we collected 7 518 185 records with 2291 different mobile phones during September and October 2019.

Based on the data of residents and suspects, we calculated the proportion of different scenario categories of two groups in Table 2. We also illustrate the importance and co-occurrence among various semantic labels in Fig. 6, where the darker areas of the pair of labels receive the more attention. For normal residents, community and hotel have a close relationship. However, for suspects, the motion patterns become more complicated, in that there are more dark areas and more co-occurrence situations, such as ⟨school, community⟩, ⟨police station, school⟩, and ⟨shopping mall, hospital⟩. From Fig. 6, it can be seen that residents and suspects have different travel motion patterns.

TABLE 2. Data statistics in different semantic categories.

Category	Normal resident	suspect
Hotel	0.2949	0.1118
Government	0.0706	0.0660
Internet cafe	0.0069	0.0010
Labor market	0.0190	0.0080
Home community	0.2605	0.2680
Supermarket	0.0357	0.0178
Gas station	0.0081	0.0064
Shopping mall	0.0478	0.1784
Police station	0.0372	0.1099
Hospital	0.0605	0.0607
Scenic spot	0.0001	0.0001
Cinema	0.0016	0.0026
School	0.1580	0.1295
Bank	0.0080	0.0398



(a) Resident (b) Suspect

FIGURE 6. Heatmaps of location concurrence.

B. EXPERIMENTAL SETUP

The platform was a Dell server 64-bit system (16-core CPU, each with 2.6 GHz, four GTX 1080ti GPUs, and 32 GB main memory). The algorithms and models were implemented in Python 3.

C. BASELINE APPROACHES

Our method is compared with a variety of competing methods grouped into three categories: classification model(CM), anomaly detection(AD) and deep learning model(DL). As the positive instances (suspects) are extremely low in our experiment, we use an under-sampling method on negative instances (residents) to balance the data in training process. All the methods are repeated 10 times in the training and validation processes with randomly selected data sets, and the averaged results are presented.

Classification Method (CM): Any supervised machine learning algorithm requires a set of informative, discriminating, and independent features. Since the movement data are too trivial, we do some preprocessing operations to extract features for classification. We generate features of trajectory from article [8], such as distance,velocity, activity scope, stop rate, HCR, VCR and curve rate for classification. After that, the classification methods, including native Bayes (NB), random forest (RF), Support Vector Machine (SVM), logistic regression (LR), Decision Tree (DTree), lightGBM (LGB), XGboost (XGB) and k-nearest-neighbor (KNN), are fitted with the training set and evaluated with the test data set.

Then, we use a two-step framework to capture the identification task. In the first step, classification methods are fitted with the training set using the above features to classify whether a trajectory belongs to a suspect. In the second step, prediction results with regard to each trajectory of the first

TABLE 3. Performance evaluations with various classifiers.

Category	Method	Result				
		Precision	Recall	F1-score	F1-micro	F1-macro
CF	RF	0.1428	0.0008	0.0016	0.9660	0.4921
	NB	0.0780	0.1750	0.1079	0.9022	0.5281
	SVM	0.0263	0.0008	0.0016	0.9651	0.4919
	LR	0.2146	0.0708	0.1065	0.9598	0.5429
	KNN	0.2017	0.0575	0.0894	0.9604	0.5346
	DTree	0.1575	0.2150	0.1818	0.9345	0.5738
	LGB	0.3764	0.2183	0.2763	0.9613	0.6282
	XGB	0.3247	0.0525	0.0903	0.9642	0.5360
AD	OCSVM	0.0538	**0.7345**	0.1003	0.5087	0.3812
	iForest	0.0066	0.0071	0.0068	0.9232	0.4834
	LOF	0.0750	0.0804	0.0776	0.9287	0.5202
DL	W2V-LSTM	0.2680	0.2492	0.2582	0.9638	0.6199
	FastText	0.1233	0.2215	0.1585	0.9405	0.5638
Ours	S2N2	0.3387	0.3023	0.3195	**0.9674**	0.6514
	$\mathrm{S2N2}_{\mathrm{semantic}}$	**0.5771**	0.2770	**0.3743**	0.9588	**0.6765**

step are utilized to predict human groups by applying the same classification method as in the previous step, like a voting mechanism.

Anomaly Detection (AD): The anomaly detection method is unsupervised and finds outliers by measuring the deviation of a given data point with its neighbors. In this work, we use One Class SVM (OCSVM), isolation forest (iForest) and Local Outlier Factor (LOF) to identify suspects, leaving only negative instances in the training set. The features of the AD methods are the same as those of the CF methods.

Deep Learning (DL): As with other techniques for human mobility prediction such as DeepMove [10] and HST-LSTM [11], which use an LSTM architecture to generate trajectories and time sections as joint features. Therefore, we generate point embedding by word to vector method, and present trajectories by LSTM models. We also adopt FastText, which is based on Facebook and is a library for efficient learning of word representations and text classification [34]. We use it to compute vector representations of points (as words) or movement paths (as text) for classification. We feed our data to FastText without time information, since we want to use this approach to examine whether the model will work when solving this problem from the viewpoint of locations alone.

D. EXPERIMENTAL RESULTS

1) PERFORMANCE COMPARISON

We adopt three widely used metrics, namely, precision, recall, and F1, as measures to evaluate the accuracy of different methods. The experimental data set is unbalanced, i.e., the number of resident instances is much larger than the number of suspects. Therefore, we also calculate the micro and macro F-scores to illustrate the effectiveness of the models. For the F1-micro value, basically the number of correctly identified predictions is divided by the total number of predictions, whereas for the F1-macro value, equal weight is given to all data categories.

2) ANALYSIS OF RESULTS

As Table 3 shows, first, AD methods are less capable than other methods, which means trajectories are diverse and difficult to cluster. And it is worth noting that the recall of OCSVM is very high, but in fact, its classification ability is not excellent, it is reasonable to believe that in the OCSVM classification process, it blindly think the predicted sample is positive which leads to this situation.

Second, the precision of human-selected features based on CF methods (except for LGB and XGB) is much lower than that of S2N2. This indicates that the movement pattern features in previous trajectory classification studies are not discriminative in our dataset.

Third, the performance of W2V-LSTM is better than that of CF methods but worse than those of LGB, XGB and S2N2. Because w2v only considers the features of the trajectory but ignores a lot of other information, the vector it generates is not good enough for classification. W2V-LSTM could be treated as another two-step method, learning features by model instead of by human definition.

Furthermore, the performance of FastText indicates that trajectory data cannot be considered simply as context. Unlike words organized in paragraphs, here, the number of distinct points is relatively small and the composition patterns of points are totally different from natural language patterns.

In the S2N2 model, we use TrajBERT to train trajectory features, and then obtain trajectory vectors for classification. Its effect is better than W2V-LSTM, which means that the method of feature extraction in TrajBERT is better than ordinary W2V. However, due to the lack of the trajectory semantic information, the effect has not improved much. In $\mathrm{S2N2}_{\mathrm{semantic}}$ model, on the basis of S2N2 model, we embed the semantic features in the vector of trajectory features and classify them based on this. The precision reached 0.5771 is significantly better than other existing methods.

Above all, End to End methods would have better performance than two-step methods with either human selected features or learned features and when trajectory classification is performed, the semantic information of the trajectory has a significant impact on the prediction results.

3) PERFORMANCE IN DIFFERENT SPECIAL CONTEXTS

To analyze the performance of various models in different semantic scenarios, we choose four semantic patterns for our

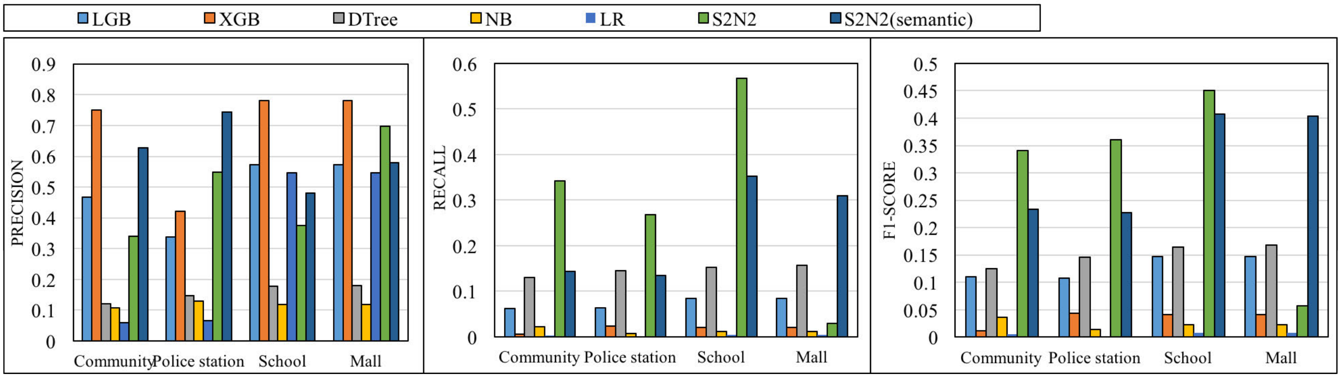


FIGURE 7. Performance in various scenarios.

experiments: community, police station, school, and shopping mall. Although the roadside has a positive impact on classification, we discard this scenario because its semantics road side are too vague.

We choose these four scenarios because they have more data and a greater impact on model output according to Table 2 and Fig. 10. We select the LGB, XGB, DTree, NB, and LR models, as well as our two models. In Fig. 7, the performance precision of S2N2 is better than that of XGB for police station, but a little poorer for the other three scenario. However, in terms of recall, our models are much better than the other methods, so S2N2 with or without semantics has much higher F1-score. From Table 3, we find that the precision and F1-score of S2N2 are improved with the use of semantic vectors in the global overview, which means that our models are more suitable for classification in complex and diverse situations.

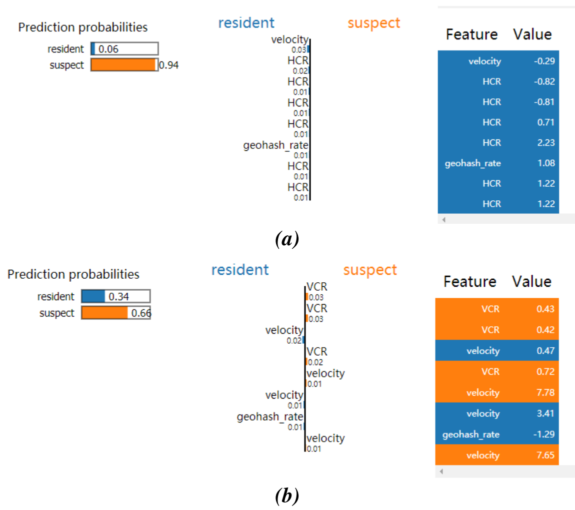


FIGURE 8. Feature selection via LIME.

VI. INTERPRETIVE MODEL AND CASE STUDY

In above sections, we have tried several different classifiers, including those based on features, sequential data, and semantic fusion models, to classify trajectories and their owners. If a machine learning model performs well, could we just trust the model and ignore the reasons for its decisions? In this section, we aim to explain users' behavior and extract more important features for understanding trajectories. Understanding the reasons behind predictions is important in assessing trust. Such understanding also provides insights into a model, which can be used to transform an untrustworthy model or prediction into a trustworthy one. We utilize popular explanation techniques, namely, LIME and SHAP, to interpret features in our model, because they have excellent performance in this regard. In more detail, we choose LIME [35] because it explains the predictions of our classifier in an interpretable and faithful manner, by learning an interpretable model locally around the prediction. We also use a unified framework for interpreting predictions, SHAP [36]. SHAP assigns each feature an importance value for a particular prediction and unifies six existing methods. These models exhibit good computational performance and consistency with human intuition.

A. CLASSIFYING TRAJECTORIES VIA FEATURES

To evaluate which features are more effective, we use the LIME tool to measure each feature for the LGB model. The visualization results are shown in Fig. 8, and we can observe that two major effective features are related to travel speed and route complexity. The velocity features represent the travel mode, such as by walking or by vehicle, the VCR (velocity change rate) represents a change in trajectory of the travel mode, and HCR (head change rate) indicates whether the travel route is complex or simple.

We also show a characteristic illustration of the entire positive case (suspects) in Fig. 9, from which we can see that only half of the suspects have obvious distinguishable travel characteristics. Therefore, it is difficult to obtain good prediction performance through a feature-based model.

B. SEMANTIC INTERPRETATION OF PLACE PREFERENCE CHOICES

Since the S2N2 model is based on the semantics of trajectories, here, we use SHAP to explain the different route choices for two groups of users. In Fig. 10, we first generate the visited location categories under user views, and then use

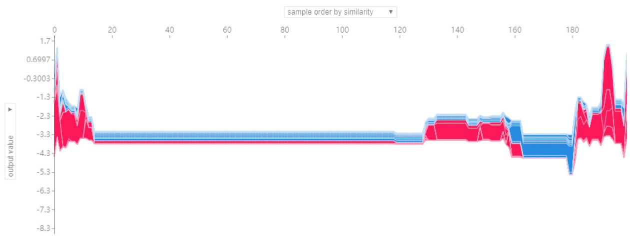


FIGURE 9. Illustration of all suspect samples.

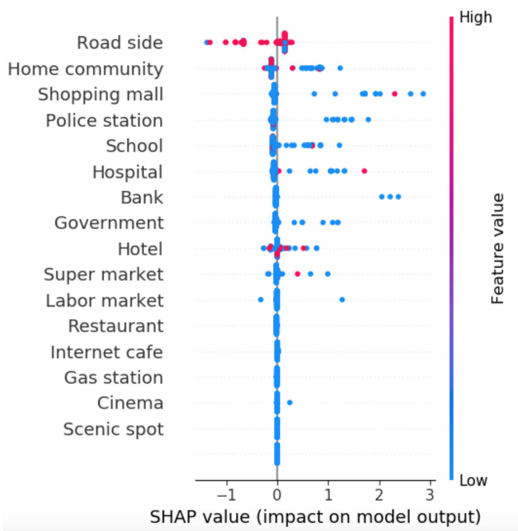


FIGURE 10. SHAP values of features for LGB.

the LGB algorithm to classify the types of users. We use the SHAP value to represent the impact of location semantics on classification. We observe that nearly half of the semantic features could help us to distinguish normal residents and suspects. In particular, the "roadside" feature is more effective in the case of positive instances, which means that, compared with ordinary residents, suspects prefer to hang out by the road, whereas residents prefer to stay at home, in the community, shopping mall, school, bank, or even the police station. Base on above observations, we select several locations as our focus scenario to be combined with trajectory embedding.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have investigated the problem of user classification from scenario views. We have proposed a framework named S2N2, which is based on a neural network and related semantics of scenarios. Since trajectory intention is difficult to understand because of travel behavior vagueness and randomness, S2N2 first generates frequent semantic patterns from a large number of trajectories, and then integrates points and semantic vectors to represent trajectories. Extensive experiments have shown that our end–end model significantly outperforms all the baselines, including classification models and anomaly detection models, on a real data set. We have also used visualization tools to present the weights of different features and explain why these features are more important for understanding and classifying user transit behavior.

There are several possible future directions for our work. First, we have only used the frequent pattern method to create semantic scenarios, and there are other approaches that could be used to describe user transit intention. Second, our current work does not consider the influence of group activities, especially for the group concurrence condition. Third, in this paper, S2N2 does not embed the time interval dimension, although this involves important features of trajectories, such as stay points. Furthermore, our classification framework is a general one, and we plan to apply it to other trajectory-based problems of interest in regional function design and public security prediction.

REFERENCES

[1] X. Ma, H. Gao, H. Xu, and M. Bian, "An IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, Dec. 2019.

[2] Y. Zhu, W. Zhang, Y. Chen, and H. Gao, "A novel approach to workload prediction using attention-based LSTM encoder-decoder network in cloud environment," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, Dec. 2019.

[3] J. Yu, C. Zhu, J. Zhang, Q. Huang, and D. Tao, "Spatial pyramid-enhanced NetVLAD with weighted triplet loss for place recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 661–674, Feb. 2020.

[4] B. Du, C. Liu, W. Zhou, Z. Hou, and H. Xiong, "Detecting pickpocket suspects from large-scale public transit records," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 3, pp. 465–478, Mar. 2019.

[5] T. Lotan and H. N. Koutsopoulos, "Approximate reasoning models for route choice behavior in the presence of information," in *Proc. 12th Int. Symp. Theory Traffic Flow Transp.*, Berkeley, CA, USA, 1993, pp. 71–88.

[6] B. Lee, A. Fujiwara, Y. Sugie, and M. Namgung, "Route choice behavior model considering randomness and vagueness uncertainty," in *Proc. 13th Mini EURO Conf. Handling Uncertainty Anal. Traffic Transp. Syst.*, Jan. 2002, pp. 64–70.

[7] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, Feb. 2010.

[8] J. Han, H. Cheng, D. Xin, and X. Yan, "Frequent pattern mining: Current status and future directions," *Data Mining Knowl. Discovery*, vol. 15, no. 1, pp. 55–86, Jul. 2007.

[9] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," *CoRR*, vol. abs/1610.00081, Oct. 2016.

[10] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin, "DeepMove: Predicting human mobility with attentional recurrent networks," in *Proc. World Wide Web Conf. (WWW)*, 2018, pp. 1459–1468.

[11] D. Kong and F. Wu, "HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction," in *Proc. IJCAI*. Palo Alto, CA, USA: AAAI Press, Jul. 2018, pp. 2341–2347.

[12] X. Jiang, E. N. de Souza, A. Pesaranghader, B. Hu, D. L. Silver, and S. Matwin, "TrajectoryNet: An embedded GPS trajectory representation for point-based classification using recurrent neural networks," in *Proc. 27th Annu. Int. Conf. Comput. Sci. Softw. Eng.*, vol. 7, 2017, pp. 192–200.

[13] H. Gong, C. Chen, E. Bialostozky, and C. T. Lawson, "A GPS/GIS method for travel mode detection in new York City," *Comput., Environ. Urban Syst.*, vol. 36, no. 2, pp. 131–139, Mar. 2012.

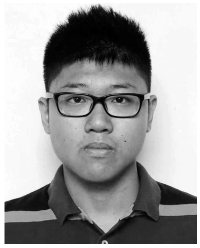
[14] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *Proc. KDD*, 2007, pp. 330–339.

[15] P. Laube and S. Imfeld, "Analyzing relative motion within groups of trackable moving point objects," in *Geographic Information Science*, M. J. Egenhofer and D. M. Mark, Eds. Berlin, Germany: Springer, 2002, pp. 132–144.

[16] M. Li, A. Ahmed, and A. J. Smola, "Inferring movement trajectories from GPS snippets," in *Proc. WSDM*, 2015, pp. 325–334.

[17] 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 *Proc. IEEE Int. Conf. Pervas. Comput. Commun. (PerCom)*, Mar. 2013, pp. 225–233.
[18] W. Luo, H. Tan, L. Chen, and L. M. Ni, "Finding time period-based most frequent path in big trajectory data," in *Proc. SIGMOD*, 2013, pp. 713–724.
[19] T. L. C. da Silva, J. A. F. de Macêdo, and M. A. Casanova, "Discovering frequent mobility patterns on moving object data," in *Proc. MobiGIS*, 2014, pp. 60–67.
[20] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and POIs," in *Proc. KDD*, 2012, pp. 186–194.
[21] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W.-Y. Ma, "Mining user similarity based on location history," in *Proc. GIS*, 2008, pp. 34:1–34:10.
[22] J. J.-C. Ying, E. H.-C. Lu, W.-C. Lee, T.-C. Weng, and V. S. Tseng, "Mining user similarity from semantic trajectories," in *Proc. LBSN*, 2010, pp. 19–26.
[23] O. Abul, F. Bonchi, and M. Nanni, "Anonymization of moving objects databases by clustering and perturbation," *Inf. Syst.*, vol. 35, no. 8, pp. 884–910, Dec. 2010.
[24] C. Zhang, K. Zhang, Q. Yuan, L. Zhang, T. Hanratty, and J. Han, "GMove: Group-level mobility modeling using geo-tagged social media," in *Proc. KDD*, 2016, pp. 1305–1314.
[25] B. Kuijpers and W. Othman, "Trajectory databases: Data models, uncertainty and complete query languages," in *Proc. ICDT*, vol. 7, 2007, pp. 224–238.
[26] O. Wolfson, B. Xu, S. Chamberlain, and L. Jiang, "Moving objects databases: Issues and solutions," in *Proc. Rafanelli*, vol. 7, 1998, pp. 111–122.
[27] Z. Li, "Semantic understanding of spatial trajectories," in *Proc. SSTD*, vol. 7, 2017, pp. 398–401.
[28] S. Brakatsoulas, D. Pfoser, and N. Tryfona, "Modeling, storing and mining moving object databases," in *Proc. IDEAS*, 2004, pp. 68–77.
[29] J. Yu, C. Hong, Y. Rui, and D. Tao, "Multitask autoencoder model for recovering human poses," *IEEE Trans. Ind. Electron.*, vol. 65, no. 6, pp. 5060–5068, Jun. 2018.
[30] J. Lee, R. M. A. Mateo, B. D. Gerardo, and S.-H. Go, "Location-aware agent using data mining for the distributed location-based services," in *Proc. ICCSA*, vol. 7, 2006, pp. 867–879.
[31] R. Song, W. Sun, B. Zheng, and Y. Zheng, "PRESS: A novel framework of trajectory compression in road networks," *Proc. VLDB Endowment*, vol. 7, no. 9, pp. 661–672, May 2014.
[32] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 328–339.
[33] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: https://arxiv.org/abs/1412.3555
[34] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 2, 2017, pp. 427–431.
[35] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. KDD*, vol. 7, 2016, pp. 77–98.
[36] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. NIPS*, vol. 7, 2017, pp. 4765–4774.

CANGHONG JIN received the Ph.D. degree in computer science from Zhejiang University, China, in 2015. Since 2016, he has been an Assistant Professor of computer science with Zhejiang University City College. He had extensive experience in big data system development, artificial intelligence applications, and software engineering management. The project he was responsible for had many successful cases in the areas of product quality supervision, public safety, public education, and other fields. His research interests include the data mining and social network analysis, and especially spatio-temporal data mining.

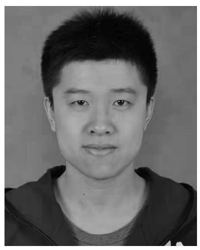


TING TAO was born in Hangzhou, Zhejiang, China, in 1999. He is currently pursuing the bachelor's degree in software engineering with the Zhejiang University City College.

In 2018, he studied big data analysis and has unique insights into special engineering. His research interests include big data and artificial intelligence. He received two Silver Prizes in the International Collegiate Programming Contest (Regional), in 2019, and the Bronze Prize in the International Collegiate Programming Contest (China Final), in 2019.

XIANZHE LUO was born in Zhejiang, China, in 1998. He received the bachelor's degree in computer technology and science from the Zhejiang University City College, in 2017. He is currently pursuing the degree.

From 2017 to 2019, he has participated at the International Collegiate Programming Contest, Asia, as a Contestant. His research interests include the implementation of various competition algorithms, and the application of deep learning in big data analysis.



ZEMIN LIU received the B.S. degree in software engineering from Shandong University, Jinan, China, in 2012, and the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2018.

He is currently a Research Scientist with Singapore Management University (SMU). Before joined SMU, he had internships with the Advanced Digital Sciences Center (ADSC) and Nanyang Technological University (NTU), in 2016 and 2017. His research had been published in conferences KDD, WWW, ICDM, and AAAI. His research interests include graph embedding, especially semantic proximity search on heterogeneous graphs, generative adversarial networks on graphs and meta-learning, including semantic proximity search, GANs-based graph embedding, and meta-learning on graph.

MINGHUI WU received the Ph.D. degree in computer science from Zhejiang University, China, in 2011. He is currently a Professor with the Zhejiang University City College. He has academic articles published in reputable journals and conferences, including AAAI, KDD, and WWW. His research interests include mobile application and artificial intelligence. He has been rewarded the Google Faculty Award three times.