Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 2860165, 12 pages
https://doi.org/10.1155/2019/2860165

Research Article

Augmented Intention Model for Next-Location Prediction from Graphical Trajectory Context

Canghong Jin,[1] **Zhiwei Lin**,[2] **and Minghui Wu**[1]

[1] *Zhejiang University City College, Huzhou Street 51, Hangzhou, Zhejiang, China*
[2] *Zhejiang University, Zheda Road 38, Hangzhou, Zhejiang, China*

Correspondence should be addressed to Minghui Wu; mhwu@zucc.edu.cn

Received 23 August 2019; Revised 27 November 2019; Accepted 6 December 2019; Published 26 December 2019

Academic Editor: Patrick Seeling

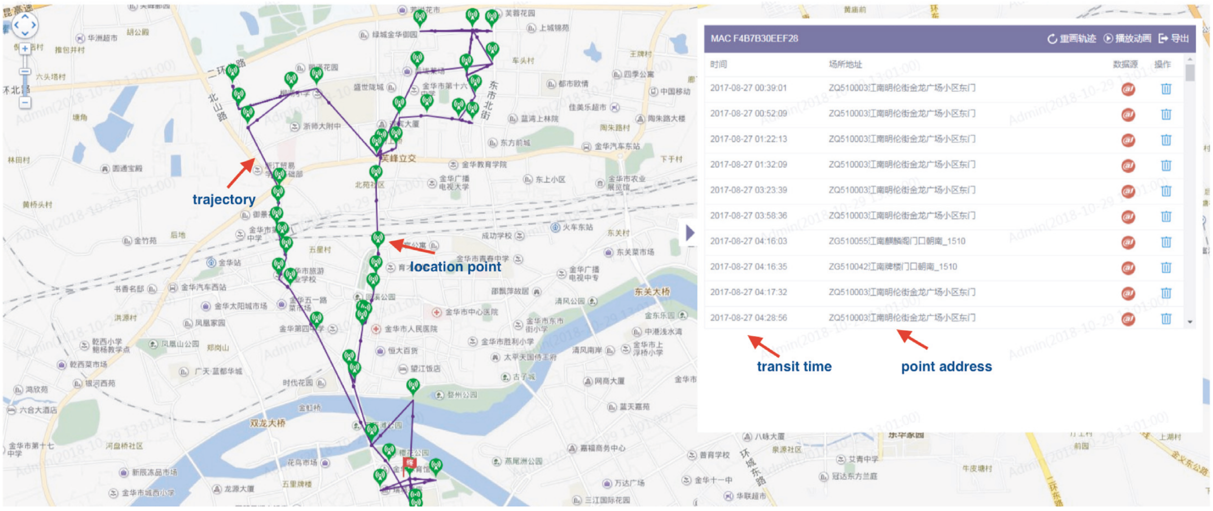
Human trajectory prediction is an essential task for various applications such as travel recommendation, location-sensitive advertisement, and traffic planning. Most existing approaches are sequential-model based and produce a prediction by mining behavior patterns. However, the effectiveness of pattern-based methods is not as good as expected in real-life conditions, such as data sparse or data missing. Moreover, due to the technical limitations of sensors or the traffic situation at the given time, people going to the same place may produce different trajectories. Even for people traveling along the same route, the observed transit records are not exactly the same. Therefore trajectories are always diverse, and extracting user intention from trajectories is difficult. In this paper, we propose an augmented-intention recurrent neural network (AI-RNN) model to predict locations in diverse trajectories. We first propose three strategies to generate graph structures to demonstrate travel context and then leverage graph convolutional networks to augment user travel intentions under graph view. Finally, we use gated recurrent units with augmented node vectors to predict human trajectories. We experiment with two representative real-life datasets and evaluate the performance of the proposed model by comparing its results with those of other state-of-the-art models. The results demonstrate that the AI-RNN model outperforms other methods in terms of top-k accuracy, especially in scenarios with low similarity.

1. Introduction

Moving records bring large opportunities and challenges for mining knowledge about public transit behavior in many location-based-service applications, such as travel recommendations, location-sensitive advertisement, and traffic planning. In contrast with other types of applications, next-location prediction is considered a difficult issue with uncomfortable and inherent uncertainties that are relevant to its unique features [1]. Transit uncertainties are mainly caused by two things: The first is *randomness* due to the nondeterministic nature of choice-behavior problems, and the second is *vagueness* uncertainty due to the lack of familiarity with road networks and the linguistic information of the network attributes [2]. Based on these two choice-behavior models and according to their study on a million users, Song et al. give the points where the movement patterns of users can easily appear random and unpredictable [3].

In particular, the dissimilarity of movement behavior in massive-scale data is an obvious problem in real-life scenarios. For example, in Figure 1, we collect user urban travel behavior by using Wi-Fi sensors in the city, thereby acquiring both geographical and temporal information. We evaluate the regularity of travel patterns of people by using similarity algorithms (e.g., Jaccard similarity) and find that the places visited by different residents differ completely in daily life. Even more, primary daily trajectories from the same user are different. Based on our experiment results, we conclude that most users do not have regular daily transit patterns—less than 20% of people have consistent patterns.

According to the large-scale observational study mentioned above, more recent developments turn to statistical- or pattern-based methods, which assume that user behavior patterns are repeatable and predictable, thereby ignoring the *randomness* and *vagueness* in real-life scenarios. Below, we list three key challenges of next-location prediction:



MAC F4B7B30EEF28 重画轨迹 播放动画 导出

时间	场所地址	数据源	操作
2017-08-27 00:39:01	ZQ510003江南明伦街金龙广场小区东门		
2017-08-27 00:52:09	ZQ510003江南明伦街金龙广场小区东门		
2017-08-27 01:22:13	ZQ510003江南明伦街金龙广场小区东门		
2017-08-27 01:32:09	ZQ510003江南明伦街金龙广场小区东门		
2017-08-27 03:23:39	ZQ510003江南明伦街金龙广场小区东门		
2017-08-27 03:58:36	ZQ510003江南明伦街金龙广场小区东门		
2017-08-27 04:16:03	ZG510055江南麒麟阁门口朝南_1510		
2017-08-27 04:16:35	ZG510042江南牌楼门口朝南_1510		
2017-08-27 04:17:32	ZQ510003江南明伦街金龙广场小区东门		
2017-08-27 04:28:56	ZQ510003江南明伦街金龙广场小区东门		

Figure 1: Urban travel records collected by Wi-Fi sensors. A trajectory is a sequential model of related movement records sorted in time series, including geography, time stamps, and MAC information.

(i) First, the quality of moving data depends on user activities and technical limitations. For example, taxi trajectories recorded by GPS devices are continuous and high precision. Although people actively post the places they visit to location-based-service applications to mark their track, these data are of low sample rate and accuracy. Another reason that user activities may be missed is because of technical limitations, e.g., a Wi-Fi sensor cannot capture mobile MAC information if it is off or if the device is in airplane mode.

(ii) Second, existing work leverages sequential models of point levels to capture movement regularities. However, mining-mobility patterns are complex and time sensitive and depend on public transport systems, activity time, weather conditions, and other possible emergencies. All of these factors interact to complicate predictions.

(iii) Third, residents living in different areas in a city have different trajectories, but the intent behind these completely different activities may be similar. Finding the underlying semantic context for each trajectory based on the points is a difficult but important task for predicting next location.

These challenges are illustrated in Figure 2, where trajectory a is an actual path with six nodes $(O_1, O_2, O_3, O_4, O_5, O_6)$. Trajectories b and c are two paths observed by the Wi-Fi sensors. For trajectory b, node O_3 is lost due to technical limitations or user activities. For trajectory c, the user went through nodes O_1 and O_5, as recorded by nearby sensors C_1 and C_2, and the observed trajectory is $(C_1, O_2, O_3, O_4, C_2, O_6)$. Therefore, the tracks b and c only present parts of actual paths with some errors, which degrades the capability of the statistical and pattern-based prediction models.

In past decades, probability models and fuzzy models have been used to deal with the problems of *randomness* and *vagueness*. Researchers now use hybrid models to interpret incomplete or discrete datasets, as exemplified by, e.g., the FLOWSIM travel-choice model [4], the LCML model [2], the AHP model [5], and the FITA model [6]. However, all of these approaches are based on predefined features, such as *distance*, *speed*, *parking conditions*, and *comfort level*, which are not available in the present study. Furthermore, neural network models perform better for predicting general locations [7, 8], and graph neural networks (GNNs) use structural interpretation for classification. Our intuition and motivation to build our model are based on these previous studies.

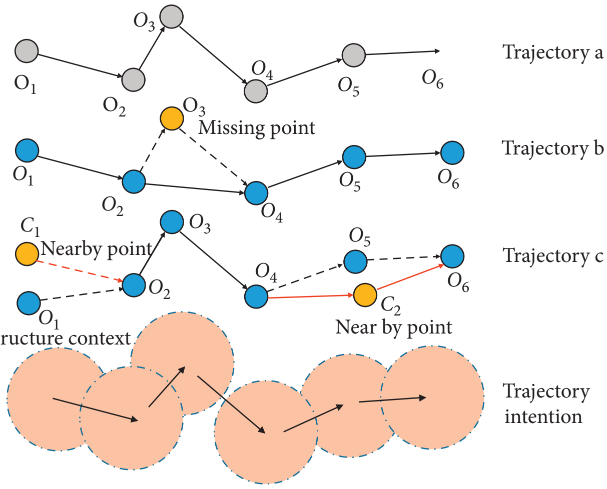


Figure 2: Problem and motivation for our model AI-RNN, where a is the ground true route, while b and c are the observed routes in different situations.

We thus propose herein an augmented-intention recurrent neural network model (AI-RNN) for location prediction in *randomness* and *vagueness* situations. In the AI-RNN model, we use a graph structure of nodes to build intent for each user through their historical transit logs in which each position node is embedded and combine the features of its adjacent nodes. Therefore, AI-RNN can describe the distinct characteristics of different locations and leverage GCN models to extend the semantic context of points. To choose suitable nodes to generate transit intention, we also propose three selection strategies to explain each trajectory for special transit scenarios. We then use a GCN to augment the vector of each point in the trajectory and use a RNN with these augmented intention points to predict the next location.

The contributions of this work are summarized as follows:

(i) We propose an AI-RNN model to capture the randomness and vagueness difficulties over a large scale of transit records. AI-RNN is thus an end-to-end trajectory-prediction approach that considers both trajectory graph structure and sequential patterns.

(ii) We design three context-selection strategies to augment user intention, including random selection, path-direction-oriented selection, and path-probability selection, which refer to various situations of user movement. We evaluate these strategies for various trajectories and determine which selection strategy works best in special scenarios.

(iii) We do extensive experiments on two real-life datasets: our urban travel dataset and the public Foursquare dataset. The results show that the AI-RNN method outperforms other statistical methods and the RNN method [7–9] for accuracy in top-k categories. Furthermore, compared with randomly selected nodes, a clear business strategy could find some nodes related to the current trajectory semantic and improve the accuracy.

The rest of the paper is organized as follows: we first formulate the problem and introduce the concepts used in the AI-RNN in Section 2. Section 3 gives the architecture of the AI-RNN and proposes three selection strategies. Next, in Section 4, we discuss experiments with two real-life datasets and evaluate the performance of the proposed model by comparing its results with those of the existing methods. Related work is discussed in Section 5, and Section 6 concludes the paper.

2. Preliminaries and Motivation

First, we formally propose the location-prediction problem under diverse trajectories and then briefly introduce the RNN gated recurrent unit (GRU) and graph convolution networks (GCNs). Finally, we discuss the motivation and present an overview of our solution.

2.1. Problem Formulation

Definition 1. A trajectory sequence is a concept of a spatial-temporal sequence generalized as $\mathscr{P} = O_1 \longrightarrow O_2 \longrightarrow \cdots \longrightarrow O_n$, where $O_i = (\mathrm{lng}_i, \mathrm{lat}_i, \mathrm{time}_i)$, $1 \le i \le n$. Trajectory sequence contains geographical information like longitude lng_i and latitude lat_i and other information about trajectory like timestamp time_i.

For given trajectories, two preprocessing steps are done: data cleaning to remove potential precision errors of Wi-Fi probes and trajectory compression to deal with redundant data acquired during data collecting.

Definition 2. The concept of regularity represents the similarity of a given user's trajectories. For example, user u has several trajectories $\mathscr{P}_1, \mathscr{P}_2, \ldots, \mathscr{P}_n$. For each trajectory $\mathscr{P}_i$, we calculate the regularity sim_i by comparing $\mathscr{P}_i$ with other trajectories via sequence similarity functions. The regularity of user u is defined as diversity $= \sum_{i=1}^{n} \mathrm{sim}_i / n$.

Definition 3. The trajectory intention is a representation of the semantic information for a trajectory. Given a trajectory $\mathscr{P}$, we can extract a set of features as this trajectory's intention. Formally, Intention = TrajectoryLength, TrajectoryComplexity, Speed, Duration Time, etc. Analyzing a trajectory intention is not only beneficial to models based on statistics but also provides valuable features as input for models and plays a significant role in understanding the potential semantics information of users.

Definition 4. Location-prediction problem: given the first few n points in a trajectory, $O = O_1, O_2, \ldots, O_n$, then for each location in the trajectory we make k prediction about top-k next location. To find out the correctness of predictions, we check whether the real next location exists in k predictions. For instance, we have a length of 10 trajectory sequence $O = O_1, O_2, \ldots, O_{10}$ and next-location prediction $O_{\mathrm{pred}_k} = O_{k1}, O_{k2}, \ldots, O_{k10}$. The true next point sequence is $O_{\mathrm{true}} = O_2, O_3, \ldots, O_{11}$, and then for each point in O_{true}, check if it exists in prediction O_{pred_k}, $k \in (1, 5, 10, 20)$.

2.2. Gated Recurrent Unit.

GRU is a gating mechanism in recurrent neural networks, introduced in 2014 by Cho et al. [10] to make each recurrent unit adaptively capture dependencies of different time scales. GRU has gating units that modulate the flow of information inside the unit. It has three major parts, which are reset gate, update gate, and activation unit. First, the reset gate r_j is computed by

$$r_j = \sigma\left(\left[W_r x\right]_j + \left[U_r h_{(t-1)}\right]_j\right). \quad (1)$$

Similarly, the update gate z_j is computed by

$$z_j = \sigma\left(\left[W_z x\right]_j + \left[U_z h_{t-1}\right]_j\right). \quad (2)$$

The activation h_t^j of the GRU at time t is a linear interpolation between the previous activation h_{t-1}^j and the candidate activation $\widehat{h}_t^j$:

$$h_t^j = \left(1 - z_t^j\right) h_{t-1}^j + z_t^j \widehat{h}_t^j. \quad (3)$$

GRU was first introduced in machine translation, which shows that this kind of model learns a semantically and syntactically meaningful representation of linguistic phrases.

Analogy to trajectory prediction, it can not only remember what happen before but also learn its internal semantics of each user's trajectories to get a better understanding.

2.3. Graph Neural Networks. GNNs have been first introduced in the study of Gori et al. [11] and modified by Scarselli et al. as a form of RNN [12]. The idea of GNNs is using the graph structure and node features to learn a representation vector of a node, improved by Li et al. [13] by introducing a modern useful strategy for RNN training to the original GNNs. This strategy is borrowed from the idea of aggregation and combination, where it iteratively updates the representation of a node by aggregating representations of its neighbors.

Spectral approaches work with a spectral representation of the graphs and have been successfully applied in the node classification problem. While Bruna et al. [14] first introduced the convolution operation is defined in the Fourier domain by computing the eigendecomposition of the graph Laplacian, it demands great computational complexity and lacks spatially localized filters. Fortunately, these issues were addressed in [15, 16]. And Kipf and Welling [17] presented the most widely used GCN model, which is designed for semisupervised learning in transductive settings, and it integrates the operations of aggregation and combination and also applies element-wise mean pooling, which successfully reduces the number of parameters and can also scale to large graphs. Formally, the model of GCN is

$$Z = f(X, A) = \mathrm{softmax}\left(\widehat{A}\mathrm{Re}LU\left(\widehat{A}XW^{(0)}\right)W^{(1)}\right). \tag{4}$$

2.4. Motivation Overview. The trajectory of an individual is a sequence of records of movements with geographical and temporal information. Thus, a suitable sequence model such as a RNN could resolve the sequential-prediction problem. However, end-to-end prediction methods depend on the quality and quantity of data. In real-life cases with diverse and sparse trajectories, the prediction accuracy would rapidly degrade. The main reason for the limitation of mobility prediction in the single-point view is that we cannot capture the complete intention of the trajectory because of the user's frequent activities or because of the technical limitations described in Figure 2. Therefore, we need to repair the context of the point and determine the user's trajectory intention before using sequential-prediction methods.

Our motivation is to find the "relevant" points of each point via the user's historical trajectory and let the target point fuse the information of the relevant points to form a fuzzy vector representation, as represented in the last chart of Figure 2. We call this process an "intent augment" and merge it into our prediction task to improve the representation.

3. Augment Intent Neural Network Framework

Figure 3 presents the architecture of AI-RNN, which comprises three key components: the intention-argument module, semantic-selection module, and sequential-prediction module.

3.1. Intention-Augment Module. Due to the *randomness* and *vagueness* of trajectories, we should merge richer semantic information to represent user intention. Here, the graph neural network-related models could be utilized to describe the relation of points by users' historical records [18].

Inspired by a relational inductive model based on graph neural networks that provide a straightforward interface for manipulating structured knowledge and producing structured behaviors [19], we design the *intention-augment module* to promote semantic of trajectories. We conduct tuple $G = (u, V, E)$, where u is the user, $V = \{v_i\}$ is the set of nodes, and each v_i is a point in historical trajectories. $E = \{(e_k, s_k, t_k)\}$ is the set of edges, where e_k is the weight of edge, s_k is the source node, and t_k is the target node.

Intention augment of a point p is a trajectory context by past visited points, which could be an edge-weighted graph. After that, GCN-based methods would fuse related points to augment the semantic of point p.

3.2. Context Generation Strategy. The critical part of AI-RNN is how to build the semantic context for each trajectory. The intention of a trajectory has two key characteristics: the time sensitiveness and the scenario sensitiveness [1]. We propose three selection strategies to represent travel semantic, including *random selection strategy*, *direction-oriented strategy*, and *maximum probability strategy*. All of these strategies could be cooperated with graph-based learning models such as [20] or [21].

3.2.1. Random Selection Strategy. The naive idea for the selection approach is to choose part of nearby points at random. Since more points merged for computing the richer semantic of a trajectory could be represented, here we could draw context between the minimum level min (i.e., the point node itself) and the maximum level max (i.e., all the neighborhood nodes). We define a threshold γ whether or not to consider a point as an augmented information by the following equation.

$$\gamma = w \times \text{random seed}(\,), \tag{5}$$

where w represents the weight of each node which is defined by the number of each node's neighborhoods, the function of random seed is a random function which outputs a number $\in [0, 1]$, and γ is a threshold, deciding whether we consider this point as an expanded branch. In this paper, we chose $\gamma = 0.6$ which provides the best result in our experiment.

Candidate points are selected from historical behaviors but should follow several criteria as shown in Algorithm 1. In Figure 4(a), we augment information of point O_1 by merging C_2 (red dashed line), as well as point O_2.

3.2.2. Direction-Oriented Strategy. This strategy is intuited by the idea that most of the people traveling in the city prefer

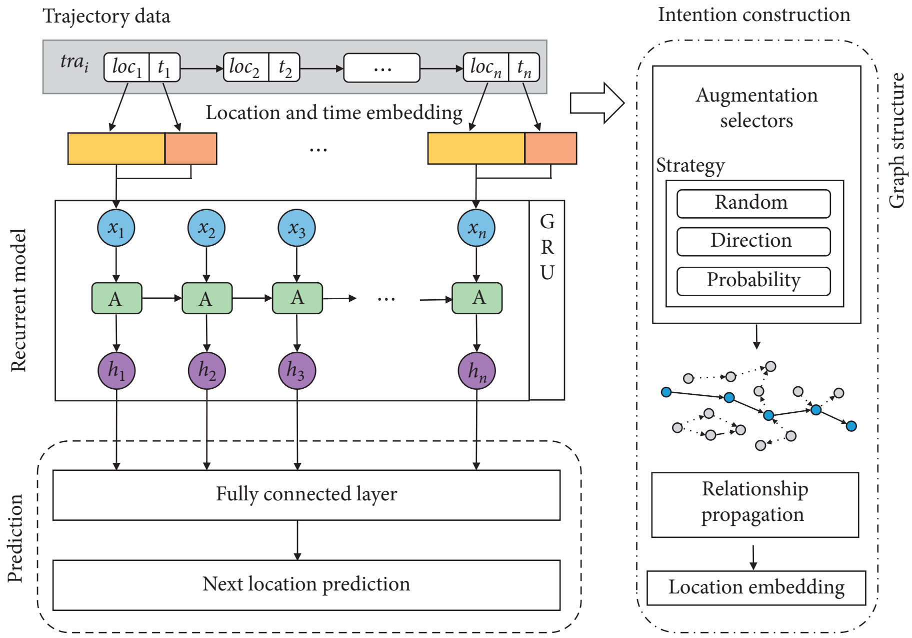


Figure 3: Model architecture.

```
Input: current trajectory $\mathscr{P}_i^u$ for user $u$
Output: A dict for candidate nodes $\mathscr{D}^u$
initialize: $\mathscr{D}^u \longleftarrow \{\}$, $time\_thr \longleftarrow 30$, $dis\_thr \longleftarrow 500$
for each node $\mathcal{O}_i \in \mathscr{P}_i^u$ do
    for each node $\mathcal{O}_j \in$ adjacent nodes of $\mathcal{N}_i$ do
        if distance between $\mathcal{O}_i$ and $\mathcal{O}_j < dis\_thr$
and $(\mathcal{O}_i(t) - \mathcal{O}_j(t)) < time\_thr$ then
            $\mathscr{D}^u[\mathcal{O}_i] \longleftarrow \mathcal{O}_j$
        end if
    end for
end for
```

Algorithm 1: Selection of candidate nodes.

a simple but straightforward path. Therefore, we could choose those nodes which keep the direction of the original path with two criteria.

First of all, the selected segments are in an individual's historical trajectories, which means the probability of transiting through these paths is relatively high. Secondly, the direction of the observed path and the additional path is similar, which means the direction difference between those two paths should be less than a threshold.

For example, as Figure 4(b) presents, given a trajectory segment $\overline{O_2O_3}$ in graph G, the direction of $\overline{O_2O_3}$, denoted by $\theta(O_2, O_3)$, is defined to be the angle of the anticlockwise rotation from the x-axis to a vector from O_2 to O_3. And we find a point C_2 from historical trajectories and get the direction of $\overline{O_2, C_2}$ as well. The "angular difference" is defined to be the minimum of the angle of clockwise and anticlockwise rotations from θ_1 to θ_2. The criterion of direction-oriented strategy is to filter those points whose angular difference between current trajectories is greater than and equal to σ:

$$\theta(O_i, O_j) = \overline{O_iO_j},$$
$$\Delta(\theta_1, \theta_2) = \min\{|\theta_1 - \theta_2|, 2\pi - |\theta_2 - \theta_1|\} < \sigma. \qquad (6)$$

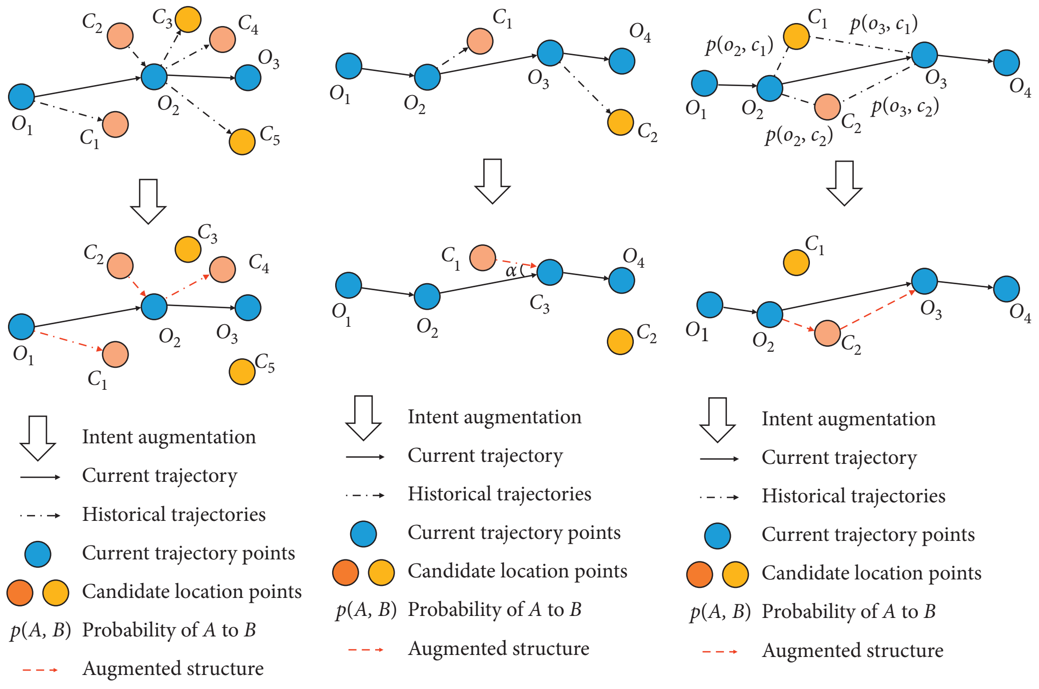
3.2.3. Maximum Probability Strategy. This model mainly demonstrates the probability that a user choosing to follow a certain path. In a given segment (O_m, C_t, O_n), O_m is a previous node, C_t is a candidate node, and O_n is the augmented point.

In maximum probability strategy, we first split whole trajectory into separated consecutive points and calculate the probability of each pair $p(O_i, O_i + 1)$. After getting all the two-point segment selection probabilities, we could calculate the three-point segment by the following equation. It should be noted that we assume that all segments are independent for simple. Of course, more complex methods could be used to calculate the selection probability of the composited trajectory:

$$p(O_m, C_t, O_n) = p(O_m, C_t) \times p(C_t, O_n). \qquad (7)$$

In reality, due to the limitation of techniques, some user traces are ignored. According to this strategy, trajectory semantics could be compensated by past behavior probabilities. For instance, as Figure 4(c) represents, for two candidate nodes (C_1 and C_2), we compare probability value of two composited segments $p(O_2, C_1, O_3)$ and $p(O_2, C_2, O_3)$. If the path through C_2 has higher confidence, we put C_2 into our augmented structure.

3.3. Training Algorithm. Algorithm 2 outlines the training process of AI-RNN. AI-RNN works in an end-to-end manner without requiring human business features. We could choose one of three strategies to augment user



C_2 C_3 C_4 O_3 O_1 O_2 C_1 C_5
C_1 O_4 O_1 O_2 O_3 C_2
C_1 $p(o_3, c_1)$ $p(o_2, c_1)$ O_1 O_2 O_3 O_4 C_2 $p(o_3, c_2)$ $p(o_2, c_2)$
C_3 α
Intent augmentation
Current trajectory
Historical trajectories
Current trajectory points
Candidate location points
$p(A, B)$ Probability of A to B
Augmented structure

(a) (b) (c)

Figure 4: Three augmentation selectors: (a) random selector, which chooses the parts of candidate nodes c_i randomly with parameter ω, (b) direction-oriented selector, which chooses the candidate nodes c_i by considering the intersection angle of paths, and (c) probability selector, which chooses c_i under maximum probability of path.

```
Input: trajectory: $\{\mathscr{P}^{u_1}, \mathscr{P}^{u_2}, \ldots, \mathscr{P}^{u_n}\}$, candidate nodes: $\{\mathscr{D}^{u_1}, \mathscr{D}^{u_2}, \ldots, \mathscr{D}^{u_n}\}$, window size: $W$
Output: trained model $\mathscr{M}$
(1)  for each user $u \in u_1, u_2, \ldots, u_n$ do
(2)    for each traj. $\mathscr{P}_i \in \mathscr{P}^u$ do
(3)      strategy 1: random selection
(4)      strategy 2: based on angle
(5)      strategy 3: based on conditional probability
(6)      //Use one of the above strategies applied on each traj.
(7)      Initialize: parameters $\theta$
(8)      for each batch nodes $W \in \mathscr{P}_i$ do
(9)        $O^u \longleftarrow \{\mathscr{P}_i(O_1), \mathscr{P}_i(O_2), \ldots, \mathscr{P}_i(O_W)\}$
(10)       $\mathscr{M} \longleftarrow$ Neural Network (GCN ({Points: $O^u$, Adj_Points: $\mathscr{D}^u$}))
(11)     end for
(12)   end for
(13) end for
```

Algorithm 2: Location prediction algorithm.

intention (lines 3–5), fusion semantic context, and utilize recurrent neural network to predict next location.

4. Performance Evaluation

4.1. Dataset. We collect two representative real-life spatiotemporal datasets to evaluate the performance of the proposed model. The first one is urban travel data (https://github.com/jincanghong/next_place_prediction) collected by Wi-Fi sensors all over the city. The second one is the public Foursquare check-in data. Since the characteristics of spatiotemporal data in these two datasets are quite different, we describe detail operations as follows.

4.1.1. Urban Travel Data. Our real-life dataset is collected by Wi-Fi sensors installed in eastern city of China, which includes MAC address, timestamp, and geo-information. We select minipart of residents and remove the private properties for our experiment. Since our study focuses on the mobility pattern of trajectories, we need to do preprocess to choose actual residents by three steps. First, according to the top-10 best-selling and most popular phone in China, we check if the MAC belongs to those android brands as follows: Huawei, OPPO, Vivo, Xiaomi, Meizu, Gionee, Samsung, Letv, and Lephone, which accounted for over 75 percent market in 2017. And then, we select those residents who have enough activity track records in total number and have at least two-week data in a month. Finally, we generate trajectories according to the criterion that distance and time interval between any two consecutive trajectory points is less than 1 kilometer and 30 minutes. We calculate the visited number n of each node and then draw the heat map with log function as shown in Figure 5.

4.1.2. Foursquare Check-In Data. This dataset contains check-ins in NYC and Tokyo collected for about 10 months (from 12 April 2012 to 16 February 2013). It contains 227,428 check-ins in New York city and 573,703 check-ins in Tokyo. Each check-in is associated with its timestamp, GPS coordinates, and semantic meaning (represented by fine-grained venue categories). Since the moving records are not continuous, we set the time interval to 3 days to generate trajectories.

Besides representing the statistic values of three datasets in Table 1, we measure the regularity of trajectories by the Jaccard function. In Figure 6, we note that most paths are short (less than 50 points), and the longer the path, the lower the value of regularity.

4.2. Experimental Setup

4.2.1. Platform. All the experiments are conducted in two environments. First one is a Cloudera platform with 24 physical machines, which is used to preprocess and generate the dataset. The other platform is a Dell server 64-bit system (16 core CPU, each with 2.6 GHz, GPU GTX 1080ti, and 32G main memory). The algorithms and models in our paper were implemented by Python 3.

4.2.2. Evaluation Criteria. We calculate precision@k to measure the performance, which means that the correct point gets to be in the top-k probabilities for it to count as "correct." Given the trajectory with n nodes, the top-k accuracy of trajectory is calculated as the average accuracy value of all nodes. The experiments are conducted in terms of the test-train model, where the first 70 percent of trajectories are chosen for training and the remaining 30 percent for testing.

4.3. Models and Performance Results. To evaluate the accuracy of our model, we compare the proposed model with some classical and state-of-the-art methods: (1) The statistic model is simple, which chooses the next node by the largest frequency value of user in the historical record. (2) The Markov model is widely used to predict location in decades, which sets up a spatiotemporal mixture model to cover the prediction task [9]. (3) The RNN model treats locations in trajectory as a sequential model and focuses on modeling the continuous spatiotemporal information. The original model as RNN works like sequence prediction, and the updated model DeepMove [7] adds historical attention to recurrent neural network, since the result has been illustrated that DeepMove has better performance than original RNN. In this experiment, we select DeepMove as state-of-the-art RNN model to compare with. We implement all the three submodels of DeepMove with different parameters as $(\max, \mathrm{dot})$ $(\mathrm{mean}, \mathrm{dot})$, and $(\max, \mathrm{concat})$. (4) AI-RNN is our proposed model that considers related external nodes to augment intention via RNN models. The first three models deal with prediction based on a series of concrete and complete nodes. Our model, on the opposite side, assumes that the observed nodes in trajectory are not complete and finds out latent intention during the prediction process.

The parameters of our model are set as follows: learning rate lr is [3e–4, 3e–3], the decay of ls is [1e–5,1e–4], embedding size of each location is 500, embedding size of time section is 48, and finally embedding size of user is 10.

4.4. Performance Comparisons. From Table 2, we have several interesting observations that confirm our research motivation. First, the RNN-based methods outperform the statistics model and the Markov model in the urban travel dataset. Our AI-RNN model is higher than DeepMove nearly 5–10 percent. In contrast, in Foursquare NY and Foursquare TKY datasets, the performance of Markov is somehow better but is still lower than those of DeepMove and AI-RNN when k is growing.

Moreover, DeepMove and AI-RNN have similar results in the Foursquare dataset, which shows that the function of augmented intention is not so effective. Observations show that if the time interval or distance in the trajectory is large, the intention will be blurred, thereby reducing the effectiveness of the AI-RNN. For AI-RNN, interestingly, the accuracy of the maximum probability strategy is better than the other two strategies in the Foursquare dataset, which demonstrates that in those LBS check-in applications, users are more likely to choose the place of interest rather than the place along the way.

4.5. Case Study and Insight Analysis. In this section, we visualize several points on the city map and illustrate the difference of semantic context in a trajectory.

Figure 7 provides examples of representative trajectory context by different strategies. Figure 7(a) represents the historical paths of the given user, in which each trajectory has a different color and the geographical dimensions are demonstrated on the city map. We can see that Figure 7(b) is the actual path $(O_1, O_2, \ldots, O_7)$ and Figure 7(c) selects nodes O_8, O_9, and O_{10} randomly which cannot draw a clear travel intention. Figure 7(d) shows that users are one way or round trip with clear directions, where points O_{11}, O_{12}, and O_{13} are merged for computing. As visualized in Figure 7(e), points O_{10} and O_{14} are generated alternate paths for prediction.

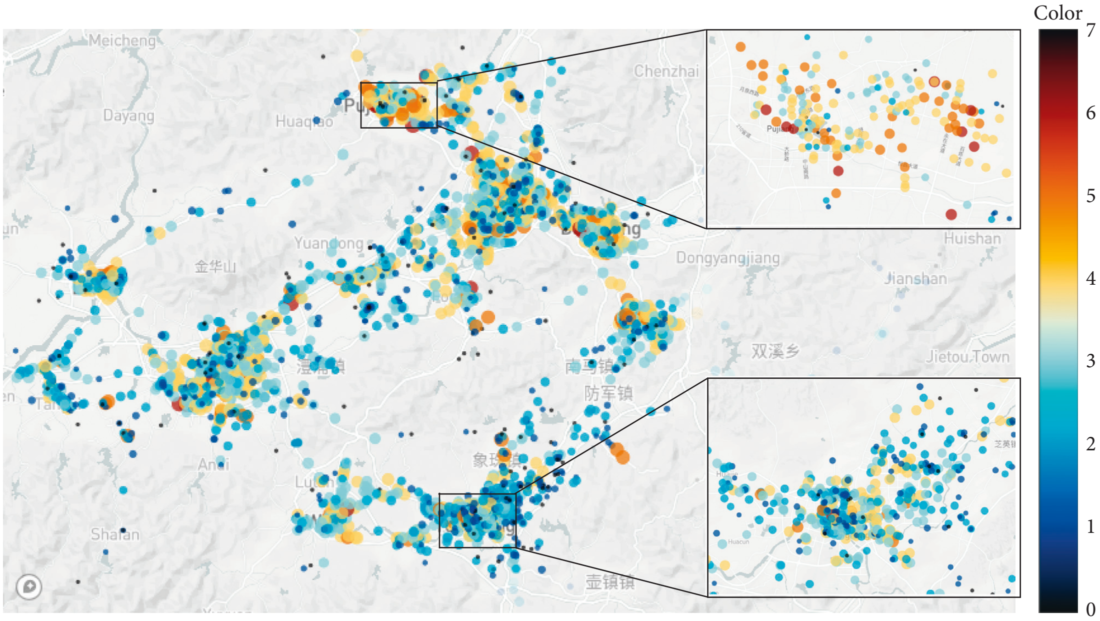
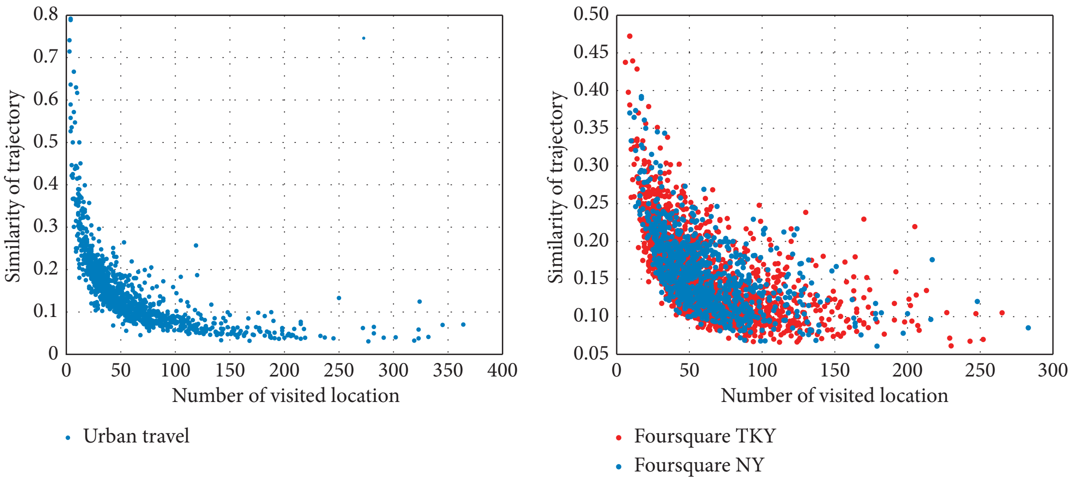


Figure 5: Heat map of visited location.

Table 1: Statistic of the dataset.

Dataset	Urban travel	Foursquare NY	Foursquare TKY
Duration	3 months	8 months	8 months
Users	1079	845	1964
Distinct points	3861	26635	40538
Trajectories	22785	7709	20397
Length of trajectory	8.60	10.74	10.87



(a)

(b)

Figure 6: Regularity of trajectories in different datasets. The x-axis is the length of trajectory and y axis-is the *Jaccard* similarity value. (a) Urban travel. (b) Foursquare.

Besides Figure 7, we further evaluate the performance under different conditions and discuss model variations via three strategies.

We first analyze the prediction result on different user groups. From Figure 6, we find that regularity of movement pattern is completely diverse in users. According to the statistical value, we select two user groups by their trajectory similarity: the Jaccard value of the regular group is larger than 0.4, while the irregular group user has the Jaccard value less than 0.2. The results are presented in Table 3, which

Table 2: Accuracy of top k results with various prediction methods.

Dataset	Urban travel				Foursquare NY				Foursquare TKY			
Methods	acc@1	acc@5	acc@10	acc@20	acc@1	acc@5	acc@10	acc@20	acc@1	acc@5	acc@10	acc@20
Statistics	22.69	36.71	39.42	41.20	9.22	17.91	21.43	25.22	12.06	21.46	24.72	28.62
Markov chain	15.42	36.52	44.32	51.48	9.78	22.24	27.08	29.10	11.76	23.68	29.61	34.53
DeepMove (max, dot)	24.94	43.79	52.06	59.60	9.35	23.62	**29.76**	34.57	10.30	24.84	**32.21**	**38.76**
DeepMove (mean, dot)	23.73	42.55	51.13	56.71	9.87	23.03	28.86	31.32	9.37	22.66	30.18	34.35
DeepMove (max, concat)	24.09	43.17	51.76	59.88	10.16	21.98	26.99	30.96	10.44	24.55	31.53	36.74
AI-RNN (RS)	27.13	51.80	62.43	71.18	12.04	23.78	27.78	32.08	11.36	23.36	28.46	33.02
AI-RNN (DO)	**27.65**	**52.42**	**62.87**	**71.95**	10.23	22.19	26.45	31.72	11.07	22.87	27.08	32.13
AI-RNN (MP)	27.14	51.89	62.19	70.90	**12.67**	**23.98**	28.02	**35.36**	**12.24**	**25.56**	32.15	35.89

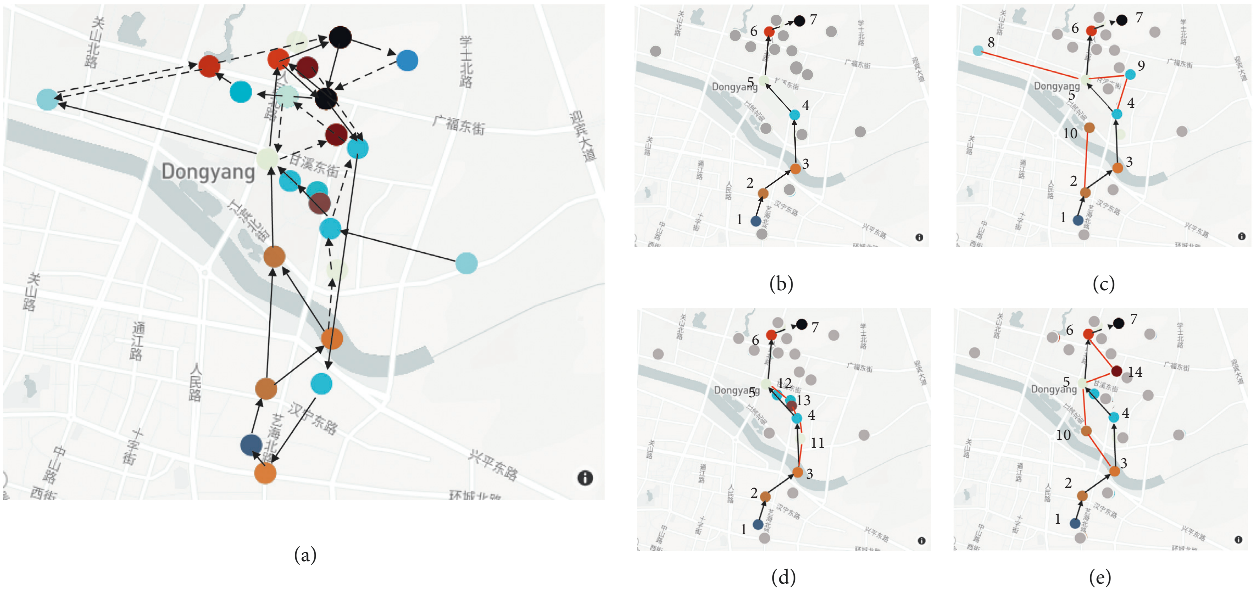


Figure 7: Case study: (a) presents the historical trajectories of the user, (b) is the actual path, and (c–e) are three separate selection strategies for prediction.

Table 3: Performance of AI-RNN in the urban travel dataset with different regularities.

Group	Strategy	acc@1	acc@5	acc@10	acc@20
Regular life	RS	73.21	94.18	96.57	97.40
Regular life	DO	**76.12**	**94.29**	**96.77**	**97.82**
Regular life	MP	74.14	94.29	96.68	97.40
Irregular life	RS	26.26	50.66	61.16	70.05
Irregular life	DO	**26.65**	**51.22**	**61.70**	**70.51**
Irregular life	MP	26.04	50.71	61.03	69.80

suggests that regular user is much easier to predict as observed in life.

Besides, we evaluate the model performance in different time sections of day. We divide trajectories into various groups by their attributes: *date*, *time*, *distance*, *duration,* and *length*. There is no difference in prediction score between the work day and holiday groups. When we move to analyze the time influence of the prediction performance as Figure 8 shows, we find that trajectories during late evening and early morning are more predictable than other time sections, but it is not easy to predict during the working hours. It means that the user transits during night time with more clear travel intention.

According to the statistic value and power law distribution shown in Figure 6, the majority length of trajectory for urban travel dataset varies from 5 to 50. Figure 9 plots the prediction accuracy obtained by all three AI-RNN models. Evaluation results demonstrate that the accuracy value varies significantly but has the same trend on all models. The performance trend is increased with the growth of the length, which is contrary to the overview of the article [7] mentioned.

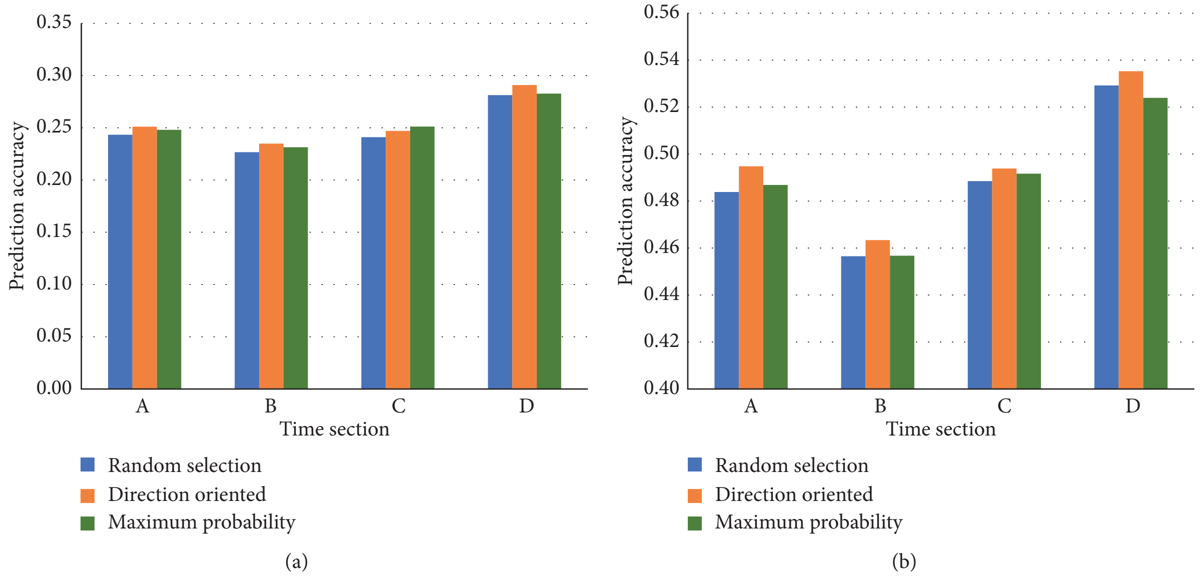


Figure 8: AI-RNN prediction results in different time sections. According to the human daily activities, we divide a day into four sections. Sections A (morning rush hour): 6 am to 9 am, B (working hours): 9 am to 16 pm, C (evening rush hours): 16 pm to 20 pm, and D (late evening and early morning): 20 pm to 6 am. (a) acc@top-1 and (b) acc@top-5.

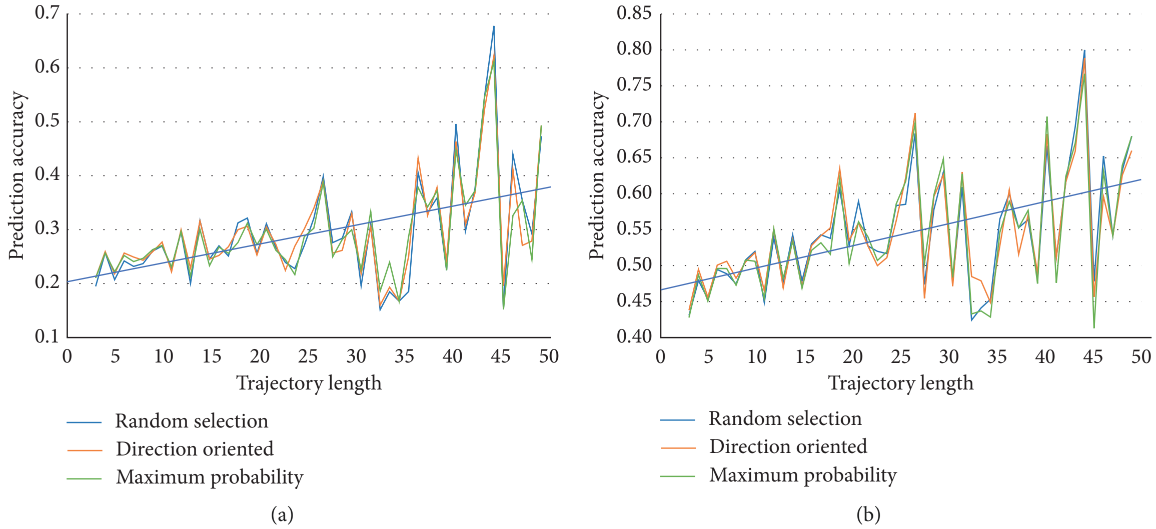


Figure 9: AI-RNN prediction results vary with trajectory length. (a) acc@top-1 and (b) acc@top-5.

5. Related Work

Works close to our task can be classified into two categories: pattern-based methods and graph-based methods. In what follows, we provide a brief overview of related work in these fields.

5.1. Pattern-Based Methods. A large number of approaches for trajectory prediction using statistics, like probability-based approach, time series, and Markov predictors, have been proposed before the rise of deep learning. Wiest et al. [22] introduced probabilistic modeling, Gaussian mixture models, which can not only get the prediction result but also learn a whole distribution over the future trajectories. Similar approaches were introduced in [23, 24]. Scellato et al. [25] introduced a spatiotemporal prediction framework based on nonlinear time series analysis [26, 27], which can estimate future times of arrival and residence times in the different significant places and where the user will be after a given time interval. Markov-based techniques have also been applied to the prediction of the destinations (geographical locations) of vehicles using, for example, partial trajectories [28]. Gambs et al. and Mathew et al. [29, 30] developed prediction location models with the Markov chains and hidden Markov models, respectively.

5.2. Relation Inductive Model. Graph-based semisupervised learning methods are used to classify the nodes and interpret the relationship between nodes [19, 31]. Moreover, since graphs are often related to other disciplines by their structures, the deep learning model for graph is critical towards decision-making problem. Researchers in articles [18, 32] summarized how to use GNNs and GCNs for relational reasoning using a unified framework called graph networks, and Lee et al. [33] reviewed the attention models for graphs. Based on the above structural learning, Yao et al. proposed a cluster-based model for location prediction, which aims at discovering groups of similar trajectories and revealing movement patterns [20]. The article [21] introduces a CNN-based approach for representing semantic trajectories and predicting future locations.

6. Conclusion

In this paper, we proposed the AI-RNN model with graph neural network and recurrent neural network. By augmenting the latent semantic of trajectory via suitable strategies, AI-RNN could improve the accuracy of next-location prediction in randomness and vagueness travel behaviors in the city. Compared with existing sequential-prediction algorithms, AI-RNN enjoys the advantage of partly utilizing the context information for every node in trajectory during the training phase. Experimental results based on real-world urban travel dataset and LBS check-in dataset show that AI-RNN outperforms the state-of-the-art baselines when the trajectories are continuous with short time interval. Additional experimental results illustrate that the duration and the distance could also make an effective influence on the performances.

Data Availability

The sample data have already been submitted and are present in GitHub. The whole dataset can be obtained via jinch@zucc.edu.cn.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The research was supported by Research Support Project of Education of Zhejiang Province (No. Y201941372).

References

[1] H. N. Koutsopoulos, T. Lotan, and Q. Yang, "A driving simulator and its application for modeling route choice in the presence of information," *Transportation Research Part C: Emerging Technologies*, vol. 2, no. 2, pp. 91–107, 1994.

[2] B. Lee, A. Fujiwara, Y. Sugie, and N. Moon, "Route choice behavior model considering randomness and vagueness uncertainty," in *Proceedings of the 13th Mini EURO Conference on Handling Uncertainty in Analysis of Traffic and Transportation Systems*, Bari, Italy, 2002.

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

[4] J. Wu, M. Brackstone, and M. Mcdonald, "Fuzzy sets and systems for a motorway microscopic simulation model," *Fuzzy Sets and Systems*, vol. 116, no. 1, pp. 65–76, 2000.

[5] T. Arslan and C. J. Khisty, "A rational reasoning method from fuzzy perceptions in route choice," *Fuzzy Sets and Systems*, vol. 150, no. 3, pp. 419–435, 2005.

[6] H. Ramazani, Y. Shafahi, and S. E. Seyedabrishami, "A fuzzy traffic assignment algorithm based on driver perceived travel time of network links," *Scientia Iranica*, vol. 18, no. 2, pp. 190–197, 2011.

[7] J. Feng, Y. Li, C. Zhang et al., "DeepMove: predicting human mobility with attentional recurrent networks," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pp. 1459–1468, International World Wide Web Conferences Steering Committee, Lyon, France, April 2018.

[8] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: a recurrent model with spatial and temporal contexts," in *Proceedings of the 2016 30th AAAI Conference on Artificial Intelligence (AAAI'16)*, pp. 194–200, AAAI Press, Phoenix, AZ, USA, February 2016.

[9] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proceedings of the 2010 International Conference on World Wide Web*, Hildesheim, Germany, April 2010.

[10] K. Cho, B. van Merrienboer, Ç. Gülçehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, http://arxiv.org/abs/1406.1078.

[11] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings of the 2005 IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 729–734, IEEE, Montreal, Canada, August 2005.

[12] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[13] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel, "Gated graph sequence neural networks," 2015, http://arxiv.org/abs/1511.05493.

[14] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2013, http://arxiv.org/abs/1312.6203.

[15] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., pp. 3844–3852, Curran Associates, Inc., New York, NY, USA, 2016.

[16] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," 2015, http://arxiv.org/abs/1506.05163.

[17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, http://arxiv.org/abs/1609.02907.

[18] A. Santoro, R. Faulkner, D. Raposo et al., "Relational recurrent neural networks," 2018, https://arxiv.org/abs/1806.01822.

[19] P. W. Battaglia, J. B. Hamrick, V. Bapst et al., "Relational inductive biases, deep learning, and graph networks," 2018, https://arxiv.org/abs/1806.01261.

[20] D. Yao, C. Zhang, Z. Zhu, J. Huang, and J. Bi, "Trajectory clustering via deep representation learning," in *Proceedings of*

the 2017 International Joint Conference on Neural Networks (IJCNN), pp. 3880–3887, IEEE, Anchorage, AK, USA, May 2017.

[21] A. Karatzoglou, N. Schnell, and M. Beigl, "A convolutional neural network approach for modeling semantic trajectories and predicting future locations," in *Proceedings of the 2018 International Conference on Artificial Neural Networks*, pp. 61–72, Springer, Rhodes, Greece, October 2018.

[22] J. Wiest, M. Höffken, U. Kreßel, and K. Dietmayer, "Probabilistic trajectory prediction with Gaussian mixture models," in *Proceedings of the 2012 IEEE Intelligent Vehicles Symposium*, pp. 141–146, IEEE, Alcala de Henares, Spain, June 2012.

[23] M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–7782, 2008.

[24] X. Liu and H. A. Karimi, "Location awareness through trajectory prediction," *Computers, Environment and Urban Systems*, vol. 30, no. 6, pp. 741–756, 2006.

[25] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell, "NextPlace: a spatio-temporal prediction framework for pervasive systems," in *Proceedings of the 9th International Conference on Pervasive Computing (Pervasive'11)*, pp. 152–169, Springer-Verlag, San Francisco, CA, USA, June 2011.

[26] H. Kantz and T. Schreiber, "Nonlinear time series analysis," *Cambridge Nonlinear Science Series*, vol. 7, Cambridge University Press, Cambridge, UK, 1997.

[27] T. Schreiber, "Efficient neighbor searching in nonlinear time series analysis," *International Journal of Bifurcation and Chaos*, vol. 5, no. 2, pp. 349–358, 1995.

[28] J. Krumm and E. Horvitz, "Predestination: inferring destinations from partial trajectories," in *Proceedings of the 2006 8th International Conference on Ubiquitous Computing (UbiComp'06)*, pp. 243–260, Springer-Verlag, Orange County, CA, USA, September 2006.

[29] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "Next place prediction using mobility Markov chains," in *Proceedings of the 1st Workshop on Measurement, Privacy, and Mobility (MPM '12)*, ACM, Bern, Switzerland, April 2012.

[30] W. Mathew, R. Raposo, and B. Martins, "Predicting future locations with hidden Markov models," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp '12)*, pp. 911–918, ACM, Pittsburgh, PA, USA, September 2012.

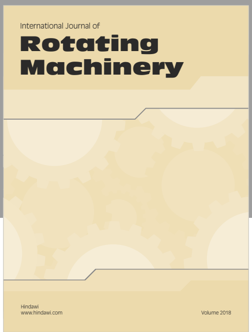
[31] J. B. Hamrick, K. R. Allen, V. Bapst et al., "Relational inductive bias for physical construction in humans and machines," 2018, https://arxiv.org/abs/1806.01203.

[32] V. Zambaldi, D. Raposo, S. Adam et al., "Relational deep reinforcement learning," 2018, https://arxiv.org/abs/1806.01830.

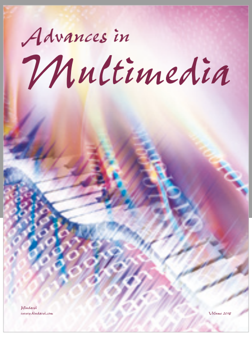
[33] J. B. Lee, R. A. Rossi, S. Kim, N. K. Ahmed, and E. Koh, "Attention models in graphs: a survey," 2018, https://arxiv.org/abs/1807.07984.

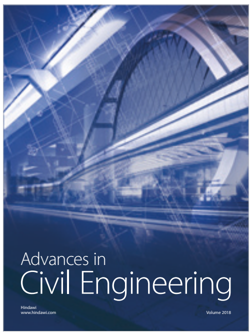
Journal of
Engineering

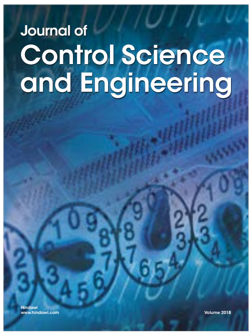
The Scientific
World Journal


International Journal of
Rotating
Machinery

Journal of
Sensors

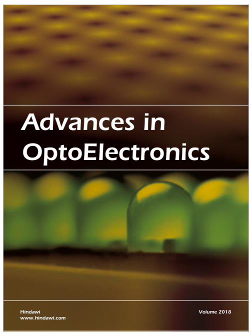

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

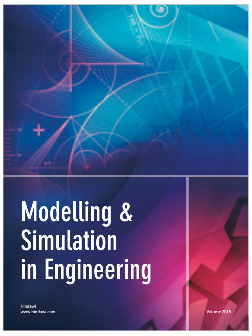
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

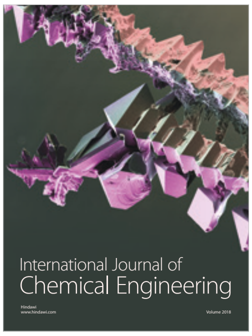

Advances in
OptoElectronics

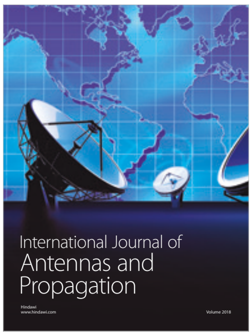
VLSI Design

International Journal of
Navigation and
Observation

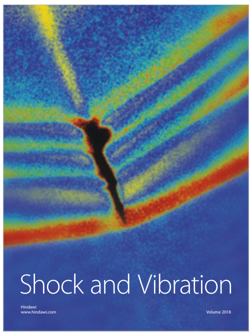

Modelling &
Simulation
in Engineering

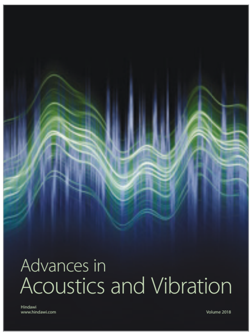
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration