

## Research Article

# A Road Network Enhanced Gate Recurrent Unit Model for Gather Prediction in Smart Cities

Mingchao Yuan <sup>1</sup>, Ling Tian <sup>1,2</sup>, Ke Yan <sup>1,2</sup> and Xu Zheng <sup>1,2</sup>

<sup>1</sup>School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

<sup>2</sup>Trusted Cloud Computing and Big Data Key Laboratory of Sichuan Province, Chengdu 610000, China

Correspondence should be addressed to Ling Tian; [lingtian@uestc.edu.cn](mailto:lingtian@uestc.edu.cn)

Received 20 April 2021; Accepted 1 July 2021; Published 15 July 2021

Academic Editor: Yan Huo

Copyright © 2021 Mingchao Yuan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Gather prediction is an indispensable part of smart city projects. The city government can respond in advance based on gather predictions and greatly reduce the loss and risks caused by vicious gatherings. Compared with other trajectory prediction tasks (i.e., the recommendation of point of interest), gather prediction pay more attention to real-time trajectory data and requests stronger spatial-temporal dependence. At the same time, gather prediction is more focused on scenes with multiple types of trajectories. And the existing methods majorly rely on the trajectory data and ignore the great influence of geographical environment (i.e., road network structure). Therefore, this paper transforms the gather prediction into the trajectory prediction task with strong real-time condition in a certain city and conducts the gathering situations by predicting users' aggregated movements in next minutes or hours. A novel Spatiotemporal Gate Recurrent Unit (STGRU) model is proposed, where spatiotemporal gates and road network gate are introduced to capture the spatiotemporal relationships between trajectories. Compared with existing methods, we improve the performance of the model by adding road network structure and external knowledges, as well as time and distance gates to reduce model parameters. The proposed STGRU is evaluated on three real-world trajectory datasets, and the experimental results demonstrate the effectiveness of the proposed model.

## 1. Introduction

In recent years, various problems caused by the gathering of people are one of the main reasons hindering urban construction and development. Crowd gatherings are prone to various accidents, such as stampede, fighting, and wounding, which place high demands on the city's management and control capabilities. Therefore, gather prediction can greatly help the city government react in advance and significantly reduce the losses and risks caused by vicious gatherings [1, 2]. As an indispensable part of smart transportation, gather prediction mainly leverages the trajectory data collected by various Internet of Things sensing devices [3–6] (such as mobile phones [7], cars, and other GPS devices [8]). These trajectory data include multiple types of patterns such as walking, driving, and public transportation. Firstly, the city government needs to predict the gathering situation to take preventive measures in advance and combine the geographical features such as rivers and buildings to divide the regions. Then, the

city government may process all the trajectory data at the current moment in the city by predicting its location at a certain time in the future. Finally, the gather is obtained through statistical methods or clustering algorithms on trajectory prediction results.

In order to achieve high-accuracy trajectory prediction, current methods majorly focus on modeling the sequential of the trajectory data and the time interval and distance interval between adjacent trajectory points. The main object is to integrate temporal and spatial features to model user behavior patterns. Typical techniques like recurrent neural networks (RNN) [9], Long short-term memory (LSTM) [10], and Gate Recurrent Unit (GRU) [11] have been successfully applied to various types of sequential data modeling and have greatly improved performance. However, none of the above methods consider time intervals and geographic informations [12] in the trajectory data. Some recent works are devoted to extending RNN and LSTM to enable modeling of time and distance intervals between neighbor points. For

example, ST-RNN [13] tries to model spatiotemporal context by extending RNN, and HST-LSTM [14] merged the spatiotemporal influence into LSTM. Recently, STGN [15] achieves SOTA by designed two pairs of time and distance gates to model the time interval and distance interval separately.

Nevertheless, trajectory data applied for gather prediction usually suffers the data sparsity [16], due to the uneven sampling interval and distribution of sensing devices. Previous efforts tried to apply spatial-temporal relations to mitigate the problem of data sparsity, but the effect is not obvious. Inspired by Li et al. [17], geographic environment information (such as road network structure) and external knowledge (such as weather information and holiday information) can effectively alleviate the problem of data sparsity. The impact of the geographic environment is essential for the modeling of short-term and long-term behavior patterns of users, and weather and holiday information will affect the overall behavior of users. For example, if the user's continuous trajectories are on the same road segment, it can be judged that the current behavior patterns are similar. Meanwhile, the long-term historical trajectory road network information can well assist in modeling the long-term behavior pattern of users. Furthermore, on weekends, more people are willing to visit more distant areas and stay in a certain location for a long time. All these side information can benefit mitigate the problem of data sparsity and improve the performance of gather prediction.

In order to make full use of external knowledge, this paper proposes a new spatiotemporal gated network by integrating road network structure and external knowledge, named Spatiotemporal Gate Recurrent Unit (STGRU). One pair of time gate and distance gate is designed to capture the short-term behavior pattern by utilize time and distance intervals, and a road network gate is introduced to memorize road network structures to model geographical environment constraints.

The proposed model abstracts the road network structure of the city into a planar graph and extracts the road network structure of a certain track point. And the weather and holiday information are integrated into the track information for input. Moreover, STGRU can model the long-term and short-term behavior patterns of users and reduce the scale of model parameters to a certain extent. Finally, the proposed model processes the trajectory prediction results with statistic methods to achieve gather prediction. Experiments show that considering the road network structure and external knowledge can effectively improve the performance of the model.

Our contributions are summarized below:

- (1) Based on the standard Gate Recurrent Unit, we proposed a Spatiotemporal Gate Recurrent Unit (STGRU) model, which integrates road network structure and external knowledge, and reduces the amount of model parameters to a certain extent
- (2) We propose an innovative gating mechanism, adding road network gate, which can model the road net-

work structure for learning spatiotemporal relationships between users' trajectories

- (3) We evaluate the proposed method on three real-world datasets including population data of Nagoya, Osaka, and Tokyo. The comprehensive comparisons with the state-of-the-art methods show the effectiveness of our model

## 2. Related Work

*2.1. Smart Transportation and Trajectory Data.* Smart transportation is a major component of smart city. The main research difficulty is the analysis and decision-making reaction of traffic information.

Traffic information analysis [18] includes traffic flow forecasting and traffic demand forecasting. Traffic flow forecasting [19, 20] and traffic congestion forecasting can help better regulate and control traffic and can effectively alleviate traffic congestion. The taxi demand forecasting method proposed by Geng et al. [21] can help taxi companies to better allocate vehicles. Li et al. [22] proposed a method for forecasting the demand for shared bicycles, which can optimize resource scheduling.

Mining and analysis of trajectory data can assist in traffic planning decisions. Wei et al. [23] used the number of stops and the parking position to analyze the effectiveness of the main line coordination.

*2.2. Spatiotemporal Data Modeling.* On the other hand, trajectory data is a type of spatiotemporal data, with two dimensions of time and space. Data mining of spatiotemporal data is very difficult, and it is also one of the current research hotspots. The Markov chain-based model [24] is a classic sequence model. And deep learning methods [25] such as RNN, LSTM, and GRU have excellent results in time modeling. The method based on matrix factorization [26] or tensor factorization [27] can model spatial features. CNN [28, 29] and GCN [30] are currently the best spatial modeling methods.

In order to capture the spatiotemporal features, Al-Molegi et al. (2016) proposed STF-RNN [31] to learn different temporal and spatial features. The TGCN [32] proposed by Zhao et al. uses GRU and GCN stacking to model spatiotemporal features. The STGCN [33] innovatively used CNN to model temporal features and achieved good results.

*2.3. Gather Prediction.* Gather prediction is an indispensable part of smart transportation, which includes many application scenarios, such as hotspot area analysis, passenger flow prediction, and population transfer prediction. Tomaharu et al. [34] proposed a collective graphical model to predict the transition populations between areas. Verma et al. [35] use trajectory data to mine hotspots and realized large-granularity gather prediction. Ni et al. [36] through passenger flow forecasting realized the gather prediction between cities. Kumar et al. [37] used trajectory clustering and similarity analysis for gather prediction. Gather prediction also can be transformed into a multitrajectory prediction task

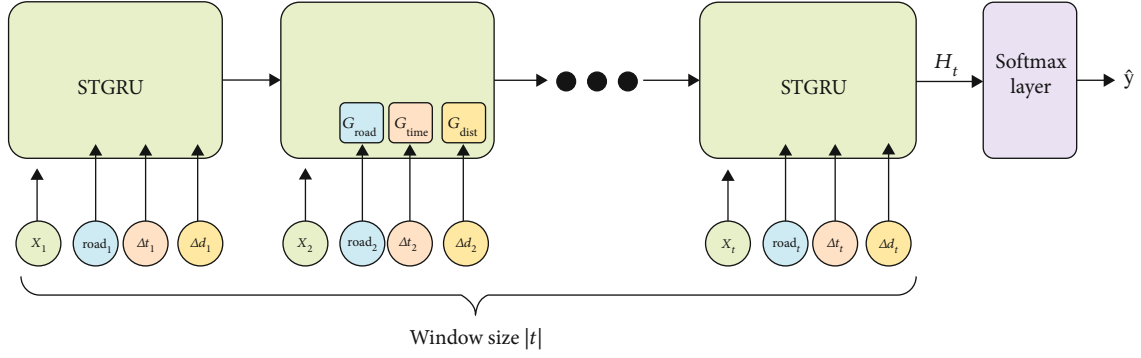


FIGURE 1: Gather prediction network model based on STGRU.

under the same time and space, which is focus on the time and space characteristics between multiple trajectories.

**2.4. Trajectory Prediction.** Different with other prediction tasks, the main features of trajectory prediction are geographic information and time information. Trajectory prediction can also use position semantics, speed, and direction. Based on the traditional probability, matrix factorization decomposes the matrix with a low-rank matrix to obtain the implicit feature vector of the user and the trajectory. Tensor factorization expands to three dimensions, including user, time information, and spatial information. Kurashima et al.'s [38] sampling is based on the subject and the distance between the user and the historical location. Liu et al. [39] combined location semantics to embed geographic context information. Research has shown that the sequence between consecutive trajectory points plays a vital role in trajectory prediction, and it is more significant in strong real-time trajectory data, because human behavior patterns are sequential. For prediction based on sequential data, the Markov chain model [40] is the most classic. Cheng et al. proposed a tensor-based model, named FPMC-LR [41], by fusing first-order Markov chains and distance constraints. Feng et al. proposed a personalized ranking metric embedding method (PRME) [42], which embeds the state at all times uniformly, and calculate the Euclidean distance between vectors to measure the similarity.

Neural networks are widely used in various tasks because they can learn to model various nonlinear features. The ST-RNN proposed by Liu et al. (2016) is the first method to introduce a deep neural network into trajectory prediction, ST-RNN uses spatiotemporal information to expand RNN, and its effect is improved. STF-RNN replaced the transition matrix with the internal representation of automatically extracted spatiotemporal features, which can more effectively discover useful features related to model human behavior. Zhu et al. [43] considered modeling time intervals to improve performance and equipped LSTM with time gating. Yang et al. [44] used neural network models to model social network structure and user trajectory behavior patterns. HST-LSTM introduces spatiotemporal factors into the gates existing in LSTM to model spatiotemporal features.

A recently proposed STGN considers the spatiotemporal context. Our proposed STGRU has the following differences from STGN. First, STGRU is extended based on GRU, which

reduces the amount of parameters and is more suitable for real-time trajectories. STGN adds time and space gates to LSTM, and the amount of parameters is more than twice that of LSTM. Secondly, STGRU is equipped with external knowledge gate to extract the road network structure to enhance the spatiotemporal characteristics and the influence of external knowledge on the overall movement pattern. However, STGN is only based on the trajectory of a single user, and it is difficult to capture the spatiotemporal relationship between users.

### 3. Method

In this section, we firstly give the definitions of gather prediction and introduce preliminaries for GRU. Then, we propose Spatio-Temporal Gated Recurrent Unit (STGRU), which uses time and distance intervals and road network structure to model short-term and long-term behavior patterns of users.

**3.1. Overview.** As shown in Figure 1, we perform trajectory prediction by stacking a STGRU layer and a softmax layer, then compare the result of trajectory prediction with the threshold  $\eta$  to obtain the result of gather prediction.

In our proposed Spatio-Temporal Gated Recurrent Unit, three gates are designed to extract spatiotemporal features and model user behavior patterns. The time gate and the distance gate can learn the time interval and distance interval in the trajectory, obtaining users' short-term behavior patterns. The road network gate aims to capture road network structural features which have the impact on short-term and long-term behavioral patterns.

In this paper, we only discuss the meshing method of dividing area, because meshing has the highest applicability. The STGRU model is also applicable to other dividing area methods, and has been echoed in comparative experiments.

**3.2. Problem Formulation.** Let  $\mathcal{U} = \{u_1, u_2, \dots, u_M\}$  be the set of  $M$  users. And according to the side length  $a$ , divide the city into a number of grids and number them, where each grid area is  $a^2$ . Each grid corresponds to a unique area ID  $r$ . For user  $u$ , she has a sequence of historical regions visited up to time  $t_{i-1}$  represented as  $H_i^u = r_{t_1}^u, r_{t_2}^u, \dots, r_{t_{i-1}}^u$ , where  $r_{t_i}^u$  means the region user  $u$  visits at time  $t_i$ .

The goal of gather prediction is to predict the regions where all users are located at time  $t_i$ . Specifically, the higher

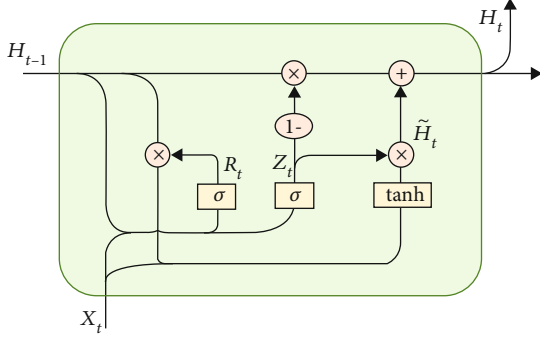
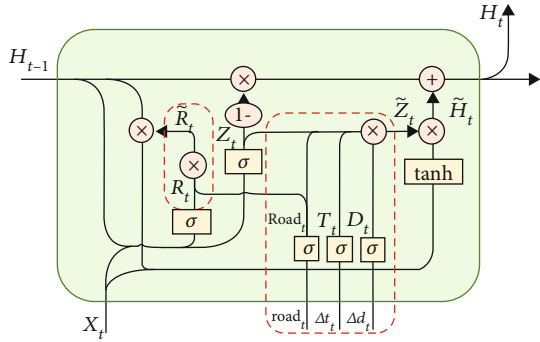


FIGURE 2: The standard GRU model.

FIGURE 3: STGRU has time gate, distance gate and road network gate, i.e.,  $T_t$ ,  $D_t$ , and  $Road_t$ .

the prediction score  $s_{r,t_i}^u$  of user  $u$  for the region  $r$  at time  $t_i$ , means the higher probability that the user  $u$  would like to located in the region  $r$  at time  $t_i$ .

According to the prediction scores of all users, predictions of  $M \times k$  possible regions can be obtained. The number of people in the region  $r$  can be obtained by counting the prediction results. And the gather in the region  $r$  can be judged whether there the number is through the threshold  $\eta$ :

$$\text{number}_{r,t_i} = \sum_{u \in U} s_{r,t_i}^u, \quad (1)$$

$$r_{\text{state}} = \begin{cases} 1, & \text{number}_{r,t_i} \geq \eta \\ 0, & \text{number}_{r,t_i} < \eta \end{cases}$$

where  $r_{\text{state}}$  is the gather state of region  $r$ , 1 means there is clustering in the region  $r$ , 0 is not, and  $\text{number}_{r,t_i}$  is the number of people in the region  $r$  at time  $t_i$ .

**3.3. Gated Recurrent Unit.** GRU (Cho et al. 2014), a variant of LSTM, can also learn the long dependency problem in RNN networks. The structure of GRU is simpler than that of the LSTM network, while the effect is very good. In order to reduce the amount of parameters to be more in line with real-time data types, our method uses standard GRU as show in Figure 2. Based on the standard LSTM network, GRU combines the forget gate and input gate of LSTM into a single update gate, removes the cell state, and uses the hidden state

to transfer information. The basic update formula of GRU is as follows:

$$R_t = \sigma(W_{xr}[H_{t-1}, X_t] + b_r), \quad (2)$$

$$Z_t = \sigma(W_{xz}[H_{t-1}, X_t] + b_z), \quad (3)$$

$$\tilde{H}_t = \tanh(W_{xh}X_t + W_{hh}(R_t \odot H_{t-1}) + b_h), \quad (4)$$

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t. \quad (5)$$

Assuming that the number of hidden units is  $h$ , the batch input  $X_t \in \mathbb{R}^{n \times d}$  at given time step is  $t$ , and the hidden state of the previous time step is  $H_{t-1}$ .  $R_t, Z_t \in \mathbb{R}^{n \times h}$  represents the reset gate and update gate, where  $\sigma(\cdot)$  is the logistic sigmoid function.  $\tilde{H}_t \in \mathbb{R}^{n \times h}$  represents the candidate hidden state at time step  $t$ , where  $\tanh(\cdot)$  is the double tangent function.  $W_{xr}, W_{xz}, W_{xh}, W_{hr}, W_{hz}, W_{hh} \in \mathbb{R}^{d \times h}$  is the weights of gates.  $b_r, b_z, b_h$  is corresponding biases. And  $\odot$  represents for the element-wise (Hadamard) product.

The reset gate  $R_t$  controls how the hidden state of the previous time step flows into the candidate hidden state of the current time step and captures long-term dependencies in the time series. The update gate  $Z_t$  can control how the hidden state should be updated by the candidate hidden state containing the current time step information and capture short-term dependencies in the time series.

**3.4. Components.** As shown in two dotted red rectangles in Figure 3, STGRU have added time gate, distance gate, and road network gate, which are denoted as  $T_t$ ,  $D_t$ , and  $Road_t$ , respectively.  $T_t$  and  $D_t$  are used to model the influence of time interval and distance interval on trajectory prediction, and  $Road_t$  is used to capture the influence of road network structure on behavior patterns. Based on GRU, time gate, distance gate, and road network gate equations are as follows:

$$T_t = \sigma(W_{xt}X_t + \sigma(\Delta t_t W_t) + b_t), \quad (6)$$

$$D_t = \sigma(W_{xd}X_t + \sigma(\Delta d_t W_d) + b_d), \quad (6)$$

$$Road_t = \sigma(W_{xroad}X_t + \sigma(\text{road}_t W_{road}) + b_{road}).$$

Combining  $T_t$ ,  $D_t$ , and  $Road_t$ , the calculation equation for reset gate and update gate is added as

$$\tilde{R}_t = R_t \odot Road_t, \quad (7)$$

$$\tilde{Z}_t = Z_t \odot T_t \odot D_t \odot Road_t,$$

then modify Eqs. (4) and (5) to

$$\tilde{H}_t = \tanh(W_{xh}X_t + W_{hh}(\tilde{R}_t \odot H_{t-1}) + b_h), \quad (8)$$

$$H_t = \tilde{Z}_t \odot H_{t-1} + (1 - \tilde{Z}_t) \odot \tilde{H}_t,$$

where  $\Delta t_t$  is the time interval and  $\Delta d_t$  is the distance interval.  $r_t$  represents road network information.  $T_t$  is equivalent to the time interval input information filter,  $D_t$  is equivalent to the distance interval input information filter, and  $Road_t$  is

used to capture the input information of the road network structure. Calculate the influence of the road network gate on the reset gate and the influence of multiple gate controls on the update gate by adding a new reset gate state  $\tilde{R}_t$  and a new update gate state  $\tilde{Z}_t$ . The influence of the gates determines the influence on the hidden state  $H_t$ .

The candidate hidden state  $\tilde{H}_t$  is determined by input information, reset gate, and the hidden state of the previous time step. The second reset gate state  $\tilde{R}_t$  is designed to memory the user's long-term road network access information.  $Road_t$  is used to memorize the road network information  $r_t$ , then transferred to  $\tilde{R}_t$ , further to  $\tilde{H}_t$ , and help simulate the long-term behavior pattern of users.

The update gate  $Z_t$  can capture short-term dependencies. Therefore, a time gate and a distance gate are designed, combined with the above road network gate to control update gate state.  $T_t$  memorizes the  $\Delta t_t$  between the track points, referring to LSTM, and uses element-wise (Hadamard) product to incorporate it into the second update gate state  $\tilde{Z}_t$ . Similarly,  $D_t$  memorizes the  $\Delta d_t$  between the track points, and  $Road_t$  memorizes the road network information and integrates it into  $\tilde{Z}_t$ . Modeling the distance interval can help capture the user's spatial behavior patterns, and modeling the time interval can help capture the user's behavior patterns such as speed and state. Modeling the road network structure can help capture users' short-term behavior constraints and long-term goals, as well as capture the spatial relationships between users.

The method of adapting the model for gather prediction is as follows. First, calculate the time interval and distance interval between track points, and  $H^u$  can be converted to

$$[(r_1^u, 0, 0), (r_2^u, t_2^u - t_1^u, d(l_1, l_2)), \dots, (r_n^u, t_n^u - t_{n-1}^u, d(l_{n-1}, l_n))]. \quad (9)$$

Secondly, add weather, holiday information, and road network information and further transform it into

$$(r_n^u, \text{weather}_n, \text{date}_n, t_n^u - t_{n-1}^u, d(l_{n-1}, l_n), \text{road}_n), \quad (10)$$

where  $r_t^u$  contains longitude, latitude, and located region;  $\text{weather}_t$  contains the highest temperature, lowest temperature, and average temperature; and  $\text{date}_t$  is marked with 0 or 1 according to whether the date is a holiday. Then,  $X_t$  in STGRU is equivalent to  $(r_t^u, \text{weather}_t, \text{date}_t)$ ,  $\Delta t_t$  is equivalent to  $t_t^u - t_{t-1}^u$ , and  $\Delta d_t$  is equivalent to  $d(l_{t-1}, l_t)$ , where  $d(\cdot, \cdot)$  is the function that computes the distance between two track points.  $\text{road}_t$  is a vector, which is concatenated by the node where the trajectory point is located in the road network graph and the neighbor nodes. For example, the road segment where the trajectory point  $i$  is located represents node  $e_i$ , and the  $c$  neighbor nodes of node  $e_i$  are, respectively, represented as  $\text{node}_{i+1}, \text{node}_{i+2}, \dots, \text{node}_{i+c}$ . Then,  $\text{road}_t$  can be expressed as  $\text{road}_t = (\text{node}_i, \text{node}_{i+1}, \dots, \text{node}_{i+c})$ . In addition, in order to extract the behavior pattern of the group, we performed a unified modeling for all users and deleted the user ID.

This paper uses a single-layer network model for comparison experiments, adds softmax layer for output, and uses the loss function of categorical cross entropy:

$$J = - \sum_{i=1}^K y_i \log(p_i). \quad (11)$$

Finally, the forecast results are counted in the same time and space. When the statistical value of a certain region  $r$  exceeds the set threshold  $\eta$ , it is considered that this region will gather.

**3.5. Analysis and Training.** STGRU reduces the amount of model parameters. The parameter quantity of a single STGRU unit can be calculated as  $3 \times d_h^2 + 6 \times d_h \times d_x + 6 \times d_h$ , where  $d_n$  is the number of hidden units, and  $d_x$  is the number of input units. Similarly, the unit parameter quantity of a LSTM unit can be calculated as  $4d_h^2 + 4 \times d_h \times d_x + 4 \times d_h$ . The GRU unit has one less gate than the LSTM unit, and the amount of parameters is also reduced accordingly, which can be calculated as  $3 \times d_h^2 + 3 \times d_h \times d_x + 3 \times d_h$ . The STGN model is the current SOTA method and is an improved model based on LSTM. The unit parameter of STGN can be calculated as  $5 \times d_h^2 + 8 \times d_h \times d_x + 10 \times d_h$ . The number of parameters of STGRU is slightly more than that of LSTM, which is one-third to one-half less than STGN.

The optimizer use Adam, a variant of Stochastic Gradient Descent (SGD), which comprehensively considers the gradient's first moment estimation (first moment estimation, the mean value of the gradient) and second moment estimation (second moment estimation), and calculates the update step length of parameters. It can automatically adjust the learning rate, and it is very suitable for large-scale data and parameter scenarios.

## 4. Experiments

In this section, we conduct experiments to evaluate the performance of our proposed model STGRU on three real-world datasets.

**4.1. Dataset.** We evaluated our proposed method on three SNS-based people flow datasets released by Nightley and Center for Spatial Information Science at the University of Tokyo (CSIS). Each dataset contains trajectories, as detailed follows.

- (i) Nagoya People Flow (NPF): NPF dataset contains the trajectories of 2387 users in the Nagoya area in 2013, with a total of 1,068,064 track points. The trajectory time is 6 days, which are July (7/22, 7/28), September (9/16, 9/22), and December (12/24, 12/29). Each trajectory starts from 0:00 to 23:55 of the current day, and the data interval is 5 minutes
- (ii) Osaka People Flow (OPF): OPF dataset contains 4924 users in the Osaka area in 2013, and the covering time is July (7/22, 7/28), September (9/16, 9/22),

and December (12/24, 12/29). The total number of track points is 2,552,883

- (iii) Tokyo People Flow (TPF): TPF dataset is the largest of the three datasets and contains the trajectories of 11536 users amount of 6,883,245 track points in the Osaka area in 2013. The time is July, 2013 (7/1, 7/7), October (10/7, 10/13), and December (12/16, 12/22).

We eliminate user data whose trajectory length was less than 30 in the three data sets and then take 70% of the users as the training set and the remaining 30% as the testing set.

The three datasets do not contain the area information of the track points. This paper divides the area of the track according to  $5\text{km} \times 5\text{km}$  in each area and determines the area where the track points are located according to the latitude and longitude of the track points in the dataset. A sliding window is used to generate samples on both training and test data, and the time interval is randomized within the sliding window to increase the complexity of the trajectory.

For example, if the time of the first track point is 8:00 am, the random interval is [7, 12, 18, 24, 30, 31]. Assuming the random number 3, then the time interval between the second track point and the first track point is  $3 \times 5 \text{ min} = 15 \text{ min}$ , which the second track point times are 8:15 am.

**4.2. Baseline Methods.** We compare our proposed model STGRU with five representative methods for trajectory prediction.

- (i) RNN [9]: it passes the state cyclically in its own network; so, it can accept a wider range of time series structure input and widely used for time series prediction tasks
- (ii) LSTM [10]: this model is suitable for processing and predicting important events with very long intervals and delays in time series. To a certain extent, the problem of gradient disappearance and gradient explosion of RNN is solved
- (iii) GRU [11]: a variant of the LSTM model, which has fewer parameters than LSTM and shows better performance on certain smaller and less frequent datasets
- (iv) HST-LSTM [14]: it integrates spatiotemporal influence into the three gates of LSTM. Since there is no session information in the datasets, its ST-LSTM vision is used here
- (v) STGN [15]: obtained by enhancing LSTM, introducing two pairs of spatiotemporal gates to capture spatiotemporal relationships

**4.3. Evaluation Metrics.** In order to evaluate the performance of our proposed STGRU model and compare it with the above five baselines, we used two standard metrics area under curve (AUC) and Recall@K. The trajectory prediction task is essentially a multiclassification task, and AUC metrics can better evaluate the classification effect. Recall@K is defined

TABLE 1: Evaluation of prediction results in terms of Recall@K and AUC on three datasets.

(a)					
NPF	Recall@1	Recall@5	Recall@10	Recall@20	AUC
LSTM	0.0428	0.1677	0.2894	0.4300	0.6951
GRU	0.0712	0.2372	0.3421	0.4771	0.7778
RNN	0.0809	0.2580	0.3570	0.4865	0.8018
ST-LSTM	0.0621	0.2438	0.3663	0.5050	0.7976
STGN	0.0762	0.2728	0.3734	0.5061	0.8177
STGRU	0.0920	0.2829	0.3891	0.5231	0.8290

(b)					
OPF	Recall@1	Recall@5	Recall@10	Recall@20	AUC
LSTM	0.0383	0.1638	0.2634	0.4375	0.7140
GRU	0.0455	0.1979	0.3007	0.4656	0.7611
RNN	0.0588	0.2258	0.3260	0.4881	0.8324
ST-LSTM	0.0502	0.2107	0.3174	0.4890	0.7817
STGN	0.0633	0.1699	0.2833	0.4920	0.8292
STGRU	0.0681	0.2439	0.3464	0.5116	0.8545

(c)					
TPF	Recall@1	Recall@5	Recall@10	Recall@20	AUC
LSTM	0.0752	0.3331	0.4849	0.6428	0.8317
GRU	0.0789	0.3319	0.4793	0.6384	0.8451
RNN	0.0856	0.3634	0.5066	0.6561	0.8645
ST-LSTM	0.0864	0.3699	0.5213	0.6682	0.8727
STGN	0.0900	0.3327	0.5103	0.6672	0.8734
STGRU	0.0933	0.3795	0.5263	0.6730	0.8755

as the ratio of the number of correct predictions to the total number of predictions. First, all possible regions are arranged in descending order according to their probability. Then, the recall score is calculated as the percentage of the number of times the true region is found among the top K most likely regions. In this paper, use  $K = 1, 5, 10, 15,$  and 20 to illustrate different results of Recall@K.  $U$  is the set of users,  $L_u$  represents the set of real regions of user  $u$  in the testing data, and  $P_{K,u}$  denotes the set of top K predicted regions; the calculation formula for Recall@K is:

$$\text{Recall@K} = \frac{1}{|U|} \sum_{u \in U} \frac{|L_u \cap P_{K,u}|}{|L_u|}. \quad (12)$$

#### 4.4. Results and Discussions

**4.4.1. Method Comparison.** Table 1 shows the performance of our proposed model STGRU and the performance of the six baselines evaluated by Recall@K and AUC on three datasets. The hidden state size is set to 32 in our experiment, the number of epochs is set to 200, and the batch size is set to 512. The

TABLE 2: The performance with different cell sizes.

Cell size	Recall@1	Recall@5	Recall@10	Recall@20
32	0.0933	0.3795	0.5263	0.6730
64	0.0922	0.3836	0.5253	0.6754
128	0.0942	0.3774	0.5217	0.6736
256	0.0923	0.3761	0.5238	0.6704
512	0.0913	0.3760	0.5190	0.6658

sliding window size is set to 10, and the random time interval in the Nagoya dataset and the Osaka dataset is 1 to 3. The random time interval of the Tokyo dataset is 1 to 5, because the data density in the Tokyo data set is higher, and needs to increase the complexity of the data by increasing the random interval. To be fair, all baseline experiments in this paper use the same hyperparameter settings.

From the experimental results, the following observations can be obtained: The STGRU model we proposed is significantly better than the existing state-of-the-art methods in all indicators of the three datasets. The performance gains provided by STGRU over these five counterparts are about 18.1%-110.2%, 5.7%-74.7%, and 3.7%-24.1% in terms of Recall@1 metric in Nagoya, Osaka, and Tokyo datasets, respectively. The results show that the mechanism of modeling the road network structure in STGRU can better model user behavior patterns, modeling short-term temporal and spatial contexts improves the effect on strong real-time data, and is effective for the task of trajectory prediction. That is because the added road network gate is combined with the update gate to integrate the short-term road network characteristics into the model, and the reset gate is combined to integrate the long-term road network characteristics.

In addition, the performance of RNN on the three datasets is better than LSTM. This is because RNN has the characteristics of short-term memory. The closer the time, the greater the weight of track points. Even if the random intervals are added, the obtained samples still have strong real-time performance; so, the performance of RNN is better. Similarly, GRU is superior to LSTM in modeling strong real-time data. The performance of HST-LSTM and STGN is better than the above three models, which proves the importance of spatiotemporal factors to track prediction. Among them, the performance of STGN is better than HST-LSTM, which proves that the method of obtaining spatiotemporal effects through specific gates is more effective than improving on the basis of LSTM gates. The reason may be the increase of the parameters.

In the three datasets, each dataset covers a total of 6 days of trajectory data in an area of Japan, and the time interval between adjacent track points is 5 minutes. Each area can be divided into about 5000 to 10000 regions. Taking the NPF dataset as an example, the number of regions is about 5000. It can be calculated that the size of the spatiotemporal matrix of the dataset is about  $5000 \times 3000$ . However, the number of trajectory points in the NPF dataset is only one million. After removing the repeated spatiotemporal regions, the size of the track point coverage matrix is less than 1% of the size of the spatiotemporal matrix of the dataset. RNN,

TABLE 3: The performance with different times and distance gates.

NPF	Recall@1	Recall@5	Recall@10	Recall@20
GRU	0.0712	0.2372	0.3421	0.4771
GRU + $D_t$	0.0887	0.2763	0.3778	0.5080
GRU + $T_t$	0.0844	0.2744	0.3773	0.5099
GRU + $D_t + T_t$	0.0909	0.2801	0.3805	0.5116

LSTM, and GRU are directly trained on the spatiotemporal matrix, which will lead to the problems of data sparsity. STGRU adds constraints between track points through time intervals, distance intervals, and road network structure. While the STGRU is being trained, only the local area covered by each sample needs to be considered, which greatly alleviates the problems of data sparsity in the dataset and can better model user behavior patterns compared to the above three models.

*4.4.2. Impact of Parameters.* In the standard RNN, different cell sizes will lead to different performance. They studied the impact of cell size on STGRU. Observe the impact of different cell sizes on model performance by changing the cell size to 32, 64, 128, 256, and 512. It can be seen from Table 2 that increasing the cell size to a certain extent can improve the performance of the model. Large cell size will increase the training time and result a decline in performance. When the number of model units is determined, the cell size determines the complexity of the model, and a larger cell size may fit the data better.

#### 4.5. Ablation Experiment

*4.5.1. Effectiveness of Time and Distance Gates.* STGRU has a time gate and a distance gate combined with update gate to capture short-term dependencies. The effectiveness of time and distance gates on modeling time and distance intervals is important. The time gate and distance gate can be closed by set  $T_t = 1$  and  $D_t = 1$ . In order to eliminate the interference of road network gates, the road network gate in STGRU was also closed. There are three sets of experiments, respectively, closing the time gate and the distance gate and closing both two gates at the same time to compare and verify the effectiveness of the time gate and the distance gate.

From Table 3, it can be found that time gate and distance gate have similar importance on the datasets. Compared with GRU, the performance improvement of  $GRU + D_t + T_t$  on the four evaluation metrics is 27.67%, 18.04%, 11.22%, and 7.23%, respectively. And the performance difference between  $GRU + T_t$  and  $GRU + D_t$  is very small, indicating that the distance interval and time interval have similar effects on modeling behavior patterns. And the performance improvement of  $GRU + D_t + T_t$  is small, indicating that there is a large degree of overlap in the characteristics of the time interval and the distance interval on the testing dataset.

*4.5.2. Effectiveness of Road Network Gates.* There is a road network gate in STGRU, which is integrated with the update gate and the reset gate to capture long-term and short-term road network dependencies. The motivation of this group is

TABLE 4: The performance with different set of road network gates.

NPF	Recall@1	Recall@5	Recall@10	Recall@20
STGRU – $R_t$	0.0909	0.2801	0.3805	0.5116
STGRU – $R_{2t}$	0.0867	0.2776	0.3786	0.5101
STGRU – $R_{1t}$	0.0862	0.2795	0.3796	0.5093
STGRU	0.0920	0.2829	0.3891	0.5231

TABLE 5: The performance with different window sizes.

Window size	Recall@1	Recall@5	Recall@10	Recall@20
10	0.0933	0.3795	0.5263	0.6730
15	0.0929	0.3897	0.5331	0.6828
20	0.0949	0.3624	0.5174	0.6786
25	0.0905	0.3794	0.5267	0.6831
30	0.1014	0.3611	0.5842	0.6957

to study the role of road network gates in the update gate and reset gate through experiments. The road network gate can be closed by setting  $Road_t = 1$  in  $\tilde{R}_t$  and  $\tilde{Z}_t$ , respectively. The effectiveness of the road network gate in capturing long-term and short-term dependencies can be verified by setting up three sets of experiments, namely, closing all road network gates and closing a single road network gate.

As shown in Table 4, the performance of closing a single road network gate is not as good as closing all road network gates. This may be due to the long-term features and short-term features of the road network structure that need to be used together. The closing of a single road network will cause the road network information to be invalid for the prediction result. At the same time, some parameters are used to model the characteristics of the road network, which will cause the performance of the model to decrease. Therefore, the performance of closing one road network gate alone is almost the same.

**4.5.3. Impact of the Sliding Window Size.** In our experiment, samples are obtained through a sliding window. The size of the sliding window limits the trajectory length of a single input. In order to compare the performance of our model in different size sliding windows, the sliding windows are set to different lengths to observe the impact, respectively, 10, 15, 20, 25, and 30. In order to ensure the number of samples under a larger sliding window size, it conducts experiments on the Tokyo People Flow dataset, because the dataset has the longest average trajectory length.

The size of the sample length determines the length of the model unit, as well as the parameters of the model. As shown in Table 5, as the length of the sliding window increases and the amount of model parameters increases, the overall performance of the model has a certain improvement. When the sliding window size is set to 30, the model complexity is 3 times that when the sliding window size is 10, the performance improvement of the four metrics increases are 8.68%, -4.85%, 11%, and 3.37%, respectively. Although the increase of the sample length can improve the performance of the model, it is necessary to consider the actually applica-

TABLE 6: The performance with different random interval sizes.

Rand interval	Recall@1	Recall@5	Recall@10	Recall@20
3	0.0889	0.3725	0.5148	0.6636
5	0.0933	0.3795	0.5263	0.6730
7	0.0920	0.3835	0.5300	0.6776
9	0.0822	0.3677	0.5196	0.6752
11	0.0747	0.3512	0.5104	0.6702

tion scenarios of the gather prediction task. The sample length within 10 is more meaningful, and this is also the main reason that the sliding window size is set to 10 in our comparison experiment with the baselines.

**4.5.4. Impact of the Random Interval Size.** Another important parameter is the size of the random interval. Increase the complexity of the trajectory samples by randomly sampling of the time interval between the track points in the sliding window. For comparison, the impact of different random intervals on the complexity of the trajectory sample and the performance of the model sets up different random intervals on the Tokyo People Flow dataset for comparison experiments and sets the random interval sizes to 3, 5, 7, 9, and 11, respectively. Choose the Tokyo People Flow dataset which the continuity in the dataset is strong, and a certain degree of complexity is required to better reflect the purpose of the experiment.

According to Table 6, it can be see that the model performance is the best when the random interval size is 5 and 7, and the random interval size that is too large and too small will cause the model performance to decrease. On the other two datasets, the model performance is better when the random interval size is 3, which is why the three data sets use different random intervals in the comparison experiment with the baselines. Using random intervals can make the sample closer to the data in the real world.

**4.6. Case Study.** The purpose of verifying the STGRU model is that it can process and predict short trajectory data and long trajectory data, which conducted two sets of experiments with the baseline models. If the user’s trajectory data is scarce, it means that it can hardly understand the user’s behavior pattern, which requires higher performance of the model. The experiments are based on the Tokyo People Flow dataset, taking data with a track length of less than 30 for calculation, without random intervals, and use recall@ $k$  as the evaluation metrics. As shown in Figure 4, STGRU has the best performance on recall@1 and recall@5, which proves that STGRU can better handle sparse data.

In another set of experiments, data with track length greater than 200 was obtained and followed the parameter settings of the comparative experiment. As shown in Figure 5, STGRU is also superior to all baselines on long trajectory data, which proves that STGRU can extract and use long-term features very well, especially the effectiveness of long-term road network features for modeling strong real-time data.



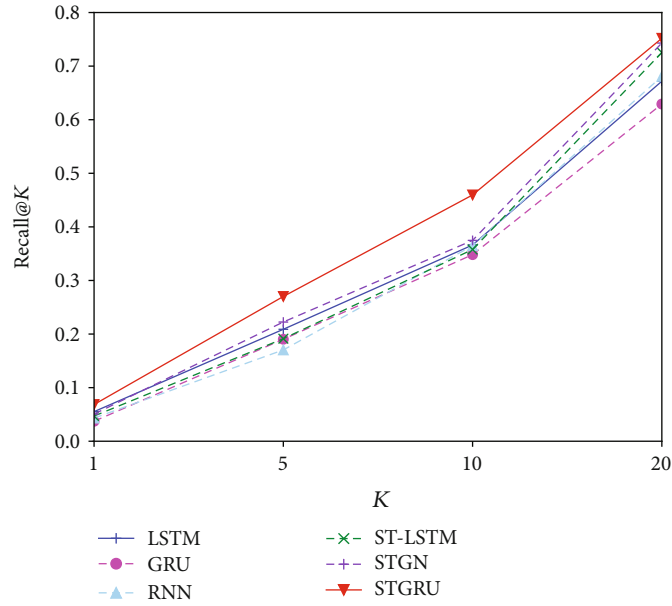


FIGURE 4: The performance on trajectory data length is less than 30.

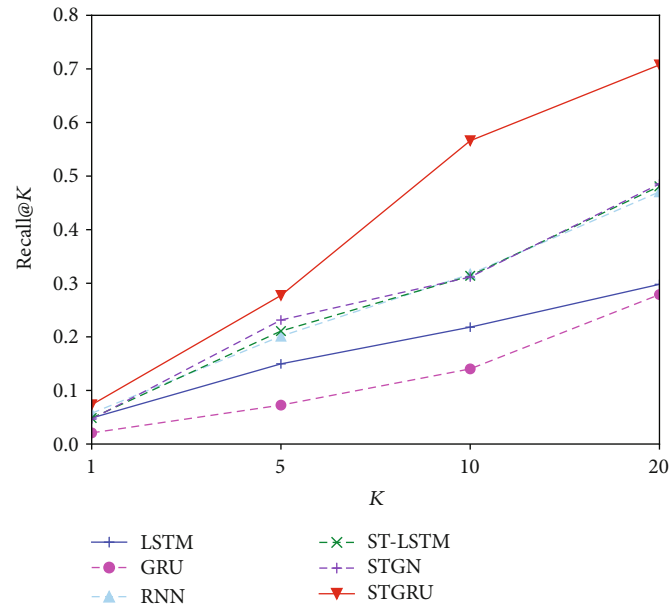


FIGURE 5: The performance on trajectory data length is greater than 200.

## 5. Conclusion

In this paper, we propose a Spatio-Temporal Gate Recurrent Unit (STGRU) model by enhancing Gate Recurrent Unit for gather prediction. In STGRU, the time gate and distance gate are introduced to model the time interval and distance interval between consecutive trajectory points, which are essential to describe the short-term behaviors of users, and the road network gate is introduced to model the long-term and short-term road network structure. We believe that the geographical environment represented by the road network

structure is very important for both the short-term and long-term behaviors of users. The three gates are combined with the update gate in the GRU to extract the user's short-term behaviors pattern. Only the road net gate and the reset gate in the GRU are combined to extract long-term behaviors patterns of users. Experimental results on three real-world datasets prove the effectiveness of our model, which is better than the latest methods.

In future work, we will further incorporate the structured representation of road network information into the model to further improve the aggregation prediction performance.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the National Key R&D Program of China (No. 2018YFC0807500), by the National Natural Science Foundation of China under Grant Nos. (U19A2059, 61802050), and by Ministry of Science and Technology of Sichuan Province Program (2018HH0075).

## References

- [1] J. Pang, Y. Huang, Z. Xie, J. Li, and Z. Cai, "Collaborative city digital twin for the covid-19 pandemic: a federated learning solution," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 759–771, 2021.
- [2] M. Sreenivasulu and M. Sridevi, "Comparative study of statistical features to detect the target event during disaster," *Big Data Mining and Analytics*, vol. 3, no. 2, pp. 121–130, 2020.
- [3] Z. Cai and Z. He, "Trading private range counting over big iot data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, July 2019.
- [4] J. Pang, Y. Huang, Z. Xie, Q. Han, and Z. Cai, "Realizing the Heterogeneity: A Self-Organized Federated Learning Framework for Iot," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3088–3098, 2020.
- [5] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart internet of things systems: a consideration from a privacy perspective," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, 2018.
- [6] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial iots," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.
- [7] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks: a survey towards private and secure applications," *ACM Computing Surveys (CSUR)*, 2021.
- [8] B. Zhao, P. Zhao, and P. Fan, "Epuf: a lightweight double identity verification in iot," *Tsinghua Science and Technology*, vol. 25, no. 5, pp. 625–635, 2020.
- [9] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*, Makuhari, Chiba, Japan, September 2010.
- [10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [11] K. Cho, B. van Merriënboer, Ç. Gülçehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014.
- [12] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2018.
- [13] 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 AAAI Conference on Artificial Intelligence*, vol. 30, Phoenix, AZ, USA, 2016.
- [14] D. Kong and F. Wu, "Hst-lstm: a hierarchical spatial-temporal long-short term memory network for location prediction," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 2341–2347, Stockholm, Sweden, 2018.
- [15] P. Zhao, A. Luo, Y. Liu et al., "Where to go next: a spatio-temporal gated network for next poi recommendation," *IEEE Transactions on Knowledge and Data Engineering*, pp. 5877–5884, 2020.
- [16] Q. Hou, M. Han, and Z. Cai, "Survey on data analysis in social media: a practical application aspect," *Big Data Mining and Analytics*, vol. 3, no. 4, pp. 259–279, 2020.
- [17] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: data-driven traffic forecasting," in *6th International Conference on Learning Representations, ICLR, Vancouver, BC, Canada*, 2018.
- [18] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6492–6499, 2019.
- [19] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: a deep learning framework for traffic prediction," *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 5668–5675, 2019.
- [20] L. Zhang, N. R. Alharbe, G. Luo, Z. Yao, and Y. Li, "A hybrid forecasting framework based on support vector regression with a modified genetic algorithm and a random forest for traffic flow prediction," *Tsinghua Science and Technology*, vol. 23, no. 4, pp. 479–492, 2018.
- [21] X. Geng, Y. Li, L. Wang et al., "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 3656–3663, 2019.
- [22] Y. Li, Z. Zhu, D. Kong, M. Xu, and Y. Zhao, "Learning heterogeneous spatial-temporal representation for bike-sharing demand prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 1004–1011, 2019.
- [23] H. Wei, G. Zheng, H. Yao, and Z. Li, "Intellilight: a reinforcement learning approach for intelligent traffic light control," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2496–2505, London, United Kingdom, 2018.
- [24] R. Begleiter, R. El-Yaniv, and G. Yona, "On prediction using variable order markov models," *Journal of Artificial Intelligence Research*, vol. 22, pp. 385–421, 2004.
- [25] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.
- [26] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," *Advances in Neural Information Processing Systems*, vol. 20, pp. 1257–1264, 2007.
- [27] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with bayesian probabilistic tensor factorization," in *Proceedings of the 2010 SIAM international conference on data mining*, pp. 211–222, Columbus, OH, USA, 2010.

- [28] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [29] R. Xin, J. Zhang, and Y. Shao, "Complex network classification with convolutional neural network," *Tsinghua Science and Technology*, vol. 25, no. 4, pp. 447–457, 2020.
- [30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017*, Toulon, France, April 2017 <http://OpenReview.net>.
- [31] A. Al-Molegi, M. Jabreel, and B. Ghaleb, "Stf-rnn: space time features-based recurrent neural network for predicting people next location," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–7, Athens, Greece, 2016.
- [32] L. Zhao, Y. Song, C. Zhang et al., "T-gcn: a temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2019.
- [33] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 3634–3640, Stockholm, Sweden, 2018.
- [34] T. Iwata and H. Shimizu, "Neural collective graphical models for estimating spatio-temporal population flow from aggregated data," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3935–3942, 2019.
- [35] N. Verma and N. Baliyan, "Pam clustering based taxi hotspot detection for informed driving," in *2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–7, Delhi, India, 2017.
- [36] M. Ni, Q. He, and J. Gao, "Forecasting the subway passenger flow under event occurrences with social media," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1623–1632, 2016.
- [37] D. Kumar, J. C. Bezdek, S. Rajasegarar, C. Leckie, and M. Palaniswami, "A visual-numeric approach to clustering and anomaly detection for trajectory data," *The Visual Computer*, vol. 33, no. 3, pp. 265–281, 2017.
- [38] T. Kurashima, T. Iwata, T. Hoshide, N. Takaya, and K. Fujimura, "Geo topic model: joint modeling of user's activity area and interests for location recommendation," in *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 375–384, Rome Italy, 2013.
- [39] Z. Liu, F. Miranda, W. Xiong, J. Yang, Q. Wang, and C. Silva, "Learning geo-contextual embeddings for commuting flow prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 808–816, 2020.
- [40] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th International Conference on World Wide Web*, pp. 811–820, North Carolina, USA, 2010.
- [41] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: successive point-of-interest recommendation," in *Twenty-Third International Joint Conference on Artificial Intelligence*, Beijing, China, 2013.
- [42] S. Feng, X. Li, Y. Zeng, G. Cong, and Y. M. Chee, "Personalized ranking metric embedding for next new poi recommendation," in *IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence*, pp. 2069–2075, Buenos Aires, Argentina, 2015.
- [43] Y. Zhu, H. Li, Y. Liao et al., "What to do next: modeling user behaviors by time-lstm," *IJCAI*, vol. 17, pp. 3602–3608, 2017.
- [44] C. Yang, M. Sun, W. X. Zhao, Z. Liu, and E. Y. Chang, "A neural network approach to jointly modeling social networks and mobile trajectories," *ACM Transactions on Information Systems (TOIS)*, vol. 35, no. 4, pp. 1–28, 2017.