IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2018.Doi Number

Multi-Context Integrated Deep Neural Network Model for Next Location Prediction

JIANXIN LIAO[1], TONGCUN LIU[1], MEILIAN LIU[2], JINGYU WANG[1], YULONG WANG[1], and HAIFENG SUN[1]

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunication, Beijing 100876 China
[2]School of Business, Guilin University of Electronic Technology, Guilin 541003 China

Corresponding author: Tongcun Liu (e-mail: liutongcun@ ebupt.com).

This work was jointly supported by (1) National Natural Science Foundation of China (No. 61771068, 61671079, 61471063, 61421061, 61372120); (2) Beijing Municipal Natural Science Foundation (No. 4152039)

ABSTRACT The prediction of next location for users in location-based social networks has become an increasing significant requirement since it can benefit both users and business. However, existing methods lack an integrated analysis of sequence context, input contexts and user preferences in a unified way, and result in an unsatisfactory prediction. Moreover, the interaction between different kinds of input contexts has not been investigated. In this paper, we propose a multi-context integrated deep neural network model (MCI-DNN) to improve the accuracy of the next location prediction. In this model, we integrate sequence context, input contexts and user preferences into a cohesive framework. Firstly, we model sequence context and interaction of different kinds of input contexts jointly by extending the recurrent neural network to capture the semantic pattern of user behaviors from check-in records. After that, we design a feedforward neural network to capture high-level user preferences from check-in data and incorporate that into MCI-DNN. To deal with different kinds of input contexts in the form of multi-field categorical, we adopt embedding representation technology to automatically learn dense feature representations of input contexts. Experimental results on two typical real-world datasets show that the proposed model outperforms the current state-of-the-art approaches by about 57.12% for Foursquare and 76.4% for Gowalla on average regarding F1-score@5.

INDEX TERMS Location-based social networks, Next location prediction, Deep neural network, Sequence prediction, Multi-context

I. INTRODUCTION

With the rapid development of wireless communication technologies and the popularization of mobile devices, the emergence of location-based social networks (LBSNs), e.g., Foursquare, Gowalla, and Yelp, has bridged the gap between cyberspace and the physical world. In LBSNs, users can post their physical locations in the form of "check-ins". They can also share their life experiences in the physical world, resulting in new opportunities to extract further insights into user preferences and behaviors [1]. Predicting location in LBSNs accurately is crucial for helping users find interesting places and services [2], for contributing to the connection of next hop in high-speed Internet of Things (IoT) [3] and for facilitating business owners to launch mobile advertisements to target users [4]. To gain significant benefit for both users and business, the prediction of next locations for users has recently attracted much academic attention [5], [6].

Predicting the next location is not just confined to estimating user preferences, which a general location prediction focuses on [7], [8]. It also includes the modeling of sequence transition from check-ins to predict user's future location. This is relevant because human movement exhibits strong sequence dependency [9], [10]. Current studies on the modeling sequence pattern are mainly based on first-order Markov Chain (MC) model such as Hidden Markov Model (HMM) [5], and Factorizing Personalized Markov Chain Model (FPMC) [9]. However, those methods are used to predict the possibility of visiting location based only on the latest check-in due to the higher computational complexity, and the influences of short-term and long-term sequence context (i.e., a set of locations visited before) have been ignored. Recently, deep neural networks have proved to be useful in modeling those sequence contexts in next location prediction. For example, by an analogous user's check-in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2827422, IEEE Access

trajectory to a sentence, Liu et al. [10] and Zhao et al. [11] employed the word2vec framework to learn the hidden representation of locations by capturing the influence of short-term sequence context. Liu et al. [13] and Yang et al. [14] leverage recurrent neural network (RNN) to capture the influence of long-term sequence context on next location decision. Cui et al. [15] propose a Hierarchical Contextual Attention-based GRU (HCA-GRU) network to capture long-term dependency and short-term interest. Their results show better performance in predicting precision than MC-based approaches.

The deep neural network has become a promising method in modeling complex sequence context. However, these methods still have some limitations. Firstly, multiple types of input contexts (e.g., time, traffic and weather condition) which generated from LBSNs have not been adequately considered to avoid the high computation cost. Those contexts have been demonstrated to be crucial for the accuracy improvement in predicting next locations for users [8]. Secondly, the interaction between different kinds of input context and its influence on users' check-in behaviors has been neglected in previous works. For instance, a user goes to cinema depending on the interaction of the time, geographic distance, and weather conditions. Thirdly, user preferences which have priority contribute to the prediction of next location, have not been well considered in those models. User preferences change with time; it is naturally determined by the locations that user visited. Current researches have proved that users are likely to have different travel schedules if they have different preferences even when the sequence trajectories and the input contexts are similar [16]. For instance, it is suitable to recommend the theatre for cinephiles, and the gym for sports fans after dinner.

Recently, a few studies focus on capturing the influence of different kinds of input contexts and sequence context in a unified manner on the prediction accuracy of next location for users. In [10]–[12], [14], [17], sequence context and input contexts were first modeled separately, and then combined by adopting an aggregation function. This method lacks a comprehensive analysis of their joint effects in a unified way. Although other studies [13], [18] incorporate input contexts into RNN by using adaptive context-specific projection matrices to model sequence context and input contexts simultaneously, these models were designed for a particular type of input context, and it is difficult to generalize them to cope with different kinds of input contexts. Moreover, the interaction of different kinds of input contexts has not been adequately investigated in the previous works.

In this paper, a novel prediction model has been established to improve the prediction performance of user's next location, called Multi-Context Integrated Deep Neural Network Model (MCI-DNN). This model is a natural extension of the deep neural network, which models sequence context, different kinds of input contexts, and user preferences in a unified framework. MCI-DNN capture semantic pattern of user behaviors from check-in data by modeling sequence context and interaction of different kinds of input contexts jointly using RNN. Subsequently, a feedforward neural network (FNN) is constructed to learn users' high-level preferences from the locations that user interacts with and then incorporate into MCI-DNN. To deal with different kinds of input contexts, embedding representation technique was adopted to automatically learn an expressive feature representation of these input contexts [19]. Compared with the traditional one-hot representation, it is less vulnerable to the data sparsity problem. Note that although our model is simple, it is more flexible and capable of capturing the joint influence of multiple context factors. It also generates a high-quality prediction.

The main contributions of this paper are summarized as follows: (1) To the best of our knowledge, it is the first time that the interaction between different kinds of input contexts was investigated to make accurate location prediction. (2) We integrate sequence context, different kinds of input contexts, and user preferences into a cohesive framework to improve the quality and capability for predicting the next location. (3) The proposed model is flexible and can incorporate other kinds of input contexts to make a prediction economically.

The remainder of this paper is organized as follows: Section 2 reviews the related works; Section 3 highlights our MCI-DNN model; Section 4 details the experimental configuration; and Section 5 depicts the experimental results and discussion, followed by the conclusion in Section 6.

II. RELATED WORK

Classical location prediction methods are based on users' check-in records and auxiliary information, such as location categories and users' social relationships, to predict where a user most likely checks-in in the future. Previous works [20]–[23] focused on the memory-based or model-based Collaborative Filtering (CF) to make location prediction. By regarding time as another dimension, Tensor Factorization (TF) based approaches were proposed to make location prediction by learning latent factors of users, items, and time bins [24]. Recently, some other works have taken different kinds of contexts into account to improve location prediction accuracy. For example, Liu et al. [8] proposed a geographical probabilistic factor model by taking geographical influences and user mobility into account. Ren et al. [25] proposed a context-aware probabilistic matrix factorization by exploiting textual information, geographical information, social information, categorical information, and popularity information. Zhou et al. [16] proposed a multi-context trajectory embedding model to systematically explore contexts.

Compared with the task of general location prediction in which the "check-in" data were considered as a whole, and their temporal relation was neglected, the essential difference in the prediction of next location is that the strong sequence dependency largely influences the performance. Chen et al. [9] first explored the dynamics of location and check-ins and proposed a personalized Markov chain model for the successive personalized location recommendation. By considering all visited locations in the check-in history of a user, the spatial-temporal sequence influence was exploited in [10] and [26]. However, these methods directly model transition probability of the observed check-in data, and fail to estimate the transition probability of the unobserved data. In addition, a metric embedding algorithm was used to compute the location transition by projecting each location into one object in a low-dimensional Euclidean latent space [6], [17]. Other works [5], [27] investigate the transition pattern of location categories to improve location prediction accuracy. These studies exploit sequence influences which are confined to first-order transitions due to data sparsity and computational complexity [10]. They cannot consider the effects of long sequence influences.

Prediction of users' next locations relies not only on the latest visited location but also on the earlier visited locations [11]. Hence, some current works [11], [12], [28] explored the influences of the location's context based on the word2vec framework by treating each location as a word and each user's visited locations as a sentence. Recently, with the successful application of deep learning on image retrieval [29], text generation [30], click prediction [10] and recommendation [31], some studies investigated the long-term sequence context influence using RNN within a whole check-in sequence. For example, Yang et al. [14] employed the RNN and GRU (Gated Recurrent Unit) models to capture the sequence relatedness in mobile trajectories at short-term or long-term. Liu et al.[32] also employed LBL (Log-Bilinear) and RNN to model short-term and long-term sequence context respectively, and then combined them into a linear model.

In the last five years, there were a few studies that hade incorporated context information into the sequence model to improve the performance of next location prediction [28]. For example, Zhao et al. [12] incorporated temporal contexts into a word2vect framework to learn location representation under some particular temporal state. The study [14] also incorporated social influence, location' context and long-term sequence dependence into a unified framework to improve the performance of next location prediction. Recently, Liu et al. [13] propose a Spatial-Temporal Recurrent Neural Networks (ST-RNN) by modeling local temporal and spatial contexts in each hidden layer with time-specific transition matrices and distance-specific transition matrices. However, these efforts towards each type of context are ad-hoc, and they limit their capacity in dealing with other kinds of contexts. Another work proposed by Liu et al. [18] adopted adaptive context-specific transition matrices to capture the external situation where user behaviors occurred. This approach is different from the conventional RNN which uses constant input and transition matrix. However, it is difficult to learn numerous parameters sufficiently due to the sparsity of check-in data. Besides, the previous studies have revealed that the interaction of different kinds of contexts has not been investigated.

III. MULTI-CONTEXT INTEGRATED MODEL

In this section, we first formulate the problem of next location prediction and then present our proposed MCI-DNN model.

A. PROBLEM DEFINITION

Definition 1 (Check-in point). A check-in point is an action conducted by a user under the specified context. For each user u, the check-in point can be denoted as a three-tuple $< l, c, t >$, where a representative user u conducts check-in action on location l under the context c at timestamp t; l is the location ID or coordinate; and c is multiple-tuple $< c_1, c_2, \dots, c_{|c|} >$ such as spatial, temporal and traffic condition.

Definition 2 (Check-in sequence). A check-in sequence is a set of check-in points with chronological order in the light of timestamp. The check-in sequence of a user u before time t_k denoted as $S_u = \{(l_1, c_1, t_1), (l_2, c_2, t_2), \dots, (l_k, c_k, t_k)\}$, k is time index.

Formally, let $U = \{u_1, u_2, \dots, u_{|U|}\}$ be a set of users and $L = \{l_1, l_2, \dots, l_{|L|}\}$ be a set of locations, where $|U|$ and $|L|$ denote the total number of unique users and locations respectively. For each user u, given a trajectory sequence S_u before time t_k, input contexts c_{k+1} at the next timestamp t_{k+1}, the task of next location prediction is to recommend top-N locations to the user u for his next move.

TABLE 1. Key Mathematical Notations.

Variable	Interpretation
U, L	The set of users and locations
S_u	The check-in sequence of user u
E_i	Embedding matrix for i-th type of input context
$c_{i,k}$	The one-hot representation vector of i-th input context in k-th type
M, W	Projection matrices for current input and last hidden layer in RNN
Q	Projection matrix for the previous layer in FNN
V	Projection Matrix for the prediction layer
h_t	Hidden state vector at time t in RNN
g^d	Represent output vector of d-th layer in FNN
RNN_t	Unit of Recurrent Neural Network (RNN) at time t
FNN	Unit of feedforward neural work (FNN)

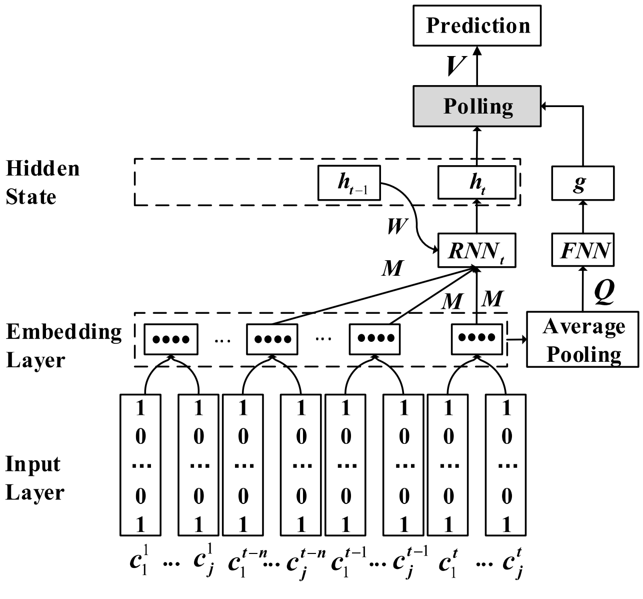


Fig.1. Architecture of the proposed Model.

B. MODEL DESCRIPTION

The key mathematical notations used in this study are shown in Table 1. Fig.1 shows the architecture of the MCI-DNN model. The arrows in Fig.1 represent data flows, c_j^t represent j-th input context encoded by one-hot representation. In our model, we first capture the semantic pattern of users from sequence context using RNN. Different kinds of input contexts are incorporated into RNN by embedding technique. Subsequently, FNN was used to capture users' high-level preferences based on the learned location latent representation from RNN and incorporated them into MCI-DNN using pooling operation to make the final prediction. The model is proposed and described in the following.

1) THE RNN MODEL FOR CONTEXT

Input contexts collected from LBSNs generally consist of multiple fields of categorical data such as location information (e.g., location ID and category ID) and temporal information (e.g., hour of the day, day of the week, and week of the month). In contrast to previous works [13], [18] that directly incorporate different kinds of input context into RNN by means of adaptive context-specific matrices, we embedded them into a dense low-dimension latent space by way of embedding representation. The entire contexts were first represented as a multi-field categorical feature vector by one-hot encoding. For example, weekday='Tuesday', the one-hot encoding can be described as [0, 1, 0, 0, 0, 0, 0]. Then, for the input feature vector $c_{i,k}$ of the i-th input context in k-th types, the embedding representation e_i is the output of the embedding layer:

$$e_i = c_{i,k} E_k \tag{1}$$

E_k is embedded matrix of k-th type of input context in embedding layer which can be learned during training. We merged the embedding vector through vector concatenation instead of element-wise product because the latter could not capture non-linear interactions between different kinds of input contexts [19]. Besides, the element-wise product requires the embedding in the same size space. Then, the value of the hidden state h_t at time t can be computed as:

$$h_t = \rho(M[e_1, e_2, \dots, e_K] + W h_{t-1}) \tag{2}$$

where K is the number of different input contexts, $\rho(\cdot)$ was a non-linear activation function, such as tanh, ReLU and sigmoid. In this study, we chose tanh as the activation function:

$$\rho(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3}$$

Additionally, the information transition from the previously hidden state is mainly determined by transition contexts between adjacent behaviors, such as time interval and geographical distance (which is a special input context). In this study, only time transition context was considered for simplification purposes, but our method can easily extend to other contexts. Instead of using continuous values of the time interval, we partitioned all the possible time intervals between two behaviors into discrete bins and then discretized them into the floor of the corresponding bin. Finally, the time-specific transition matrix was utilized. Then, equation (2) could be rewritten as:

$$h_t = \rho(M[e_1, e_2, \dots, e_K] + W_{c_{t-1}} h_{t-1}) \tag{4}$$

where $W_{c_{t-1}}$ is projection matrix for h_{t-1} under transition context c_{t-1}.

The basic RNN model assumes that the temporal dependency changes monotonously along with positions in a sequence by only modeling one element in each hidden layer. Such an assumption is reasonable in modeling words in a sentence or frames in a video, as adjacent words or frames have significant correlation [18]. However, it is unsuitable for modeling the complex human mobility in a real situation because users usually complete successive check-in numerous of locations in a short time [3, 8]. Hence, previous check-in behaviors usually have a strong impact on the current and future decision [18]. To model such short-term sequence contexts, we further extend RNN to model multiple elements in each hidden layer to capture location relatedness context, as is shown in Fig.1. Finally, the equation (4) could be rewritten as:

$$h_t = \rho(\sum_{j=1}^{n-1} M[e_1, e_2, \dots, e_K] + W_{c_{t-1}} h_{t-1}) \tag{5}$$

where n is the number of elements modeled in a sequence, which is also called slide window width. In particular, Equation (5) will be changed into basic RNN if and only if one input element was considered in each hidden layer (i.e., $n = 1$).

2) THE FNN MODEL FOR PREFERENCE

Naturally, user preferences are determined by the locations he or she visited. Inspired by [17], [33], we built a feedforward neural network (FNN) to learn high-level latent preference automatically. For simplicity, all users shared the same size of neurons in the hidden layers, which also helped us identify the common check-in patterns of users. Given the d-1-th hidden layer g^{d-1}, the state of d-th is updated as:

$$g^d = \sigma(Q^d g^{d-1}) \tag{6}$$

where Q denotes projection matrix for the previous layer as input, $\sigma(x) := 1/1 + e^{-x}$ is the logistic sigmoid function.

We regarded each trajectory sequence as an ordered list of input elementsd $= (e_{l_1}, e_{l_2}, \dots, e_{l_t}, \dots)$, where e_{l_t} is learned embedding vector of the location visited by user u at time t according to equation (1). Thus, the input vector g^0 could be represented by a weighted average of all visited elements in the sequence, computed as:

$$g^0 = \frac{\sum_{t=1}^{|S_u|} w(t) e_{l_t}}{\sum_{t=1}^{|S_u|} w(t)} \tag{7}$$

where $w(t)$ denotes the degree of user preferences on location l_t, which can be any weighting function. We used TF-IDF as the weighting function. $|S_u|$ denotes the length of check-in sequence.

C. PARAMETER LEARNING

The prediction of user u visiting location l at time $t+1$ can be influenced by semantic pattern learned from check-in and users' personal preferences. Hence, the prediction function can be written as:

$$r_{u,l}^{t+1} = h_t \oplus g_u V \tag{8}$$

where ⊕ denote pooling operation.

The neural network is usually trained by back propagation (BP) algorithm in natural language processing. In this study, only a small number of the visited locations were recorded in LBSNs. The density of the check-in data used for location prediction is approximate 0.1%. This value is largely sparse compared with a traditional recommendation task, such as movie recommendation. The unrecorded locations may be either negative feedback or unknown for users. Similar to [13], we trained our model by Bayesian analysis of Personalized Ranking (BPR) criteria [34] and Back Propagation Through Time (BPTT) algorithm to learn the parameters of the proposed model.

The BPR algorithm is a matrix factorization method that uses pairwise ranking loss. The basic assumption of BPR is that a user prefers the selected items to unselected ones. This method has been used successfully for parameter estimation of RNN-based recommendation models [13]. In the BPR algorithm, given a visited location l and sampled negative location l', the pairwise preference probability can be given as:

$$p(l > l'; u, t+1 | \theta) = \sigma(r_{u,l}^{t+1} - r_{u,l'}^{t+1}) \tag{9}$$

where $\theta = (E, M, W, V, Q)$ denotes all parameters to be learned. Then, incorporating the negative log-likelihood function, we have the final objective function:

$$\mathcal{J} = -\sum \ln \sigma(r_{u,l}^{t+1} - r_{u,l'}^{t+1}) + \frac{\lambda}{2}\|\theta\|^2 \tag{10}$$

where λ is the regularization parameter to avoid overfitting. According to Equation (10), derivations of $\mathcal{J}$ with respect to the parameters θ can be calculated.

Moreover, parameters in RNN can be further learned by using BPTT algorithm. According to Equation (5), given the derivation $\delta_{\mathcal{J}}/h_t$, the corresponding gradients of all parameters in the hidden layer can be calculated. Moreover, we adopted a dropout technique to avoid overfitting. In our work, we simply set a fixed drop ratio (50%) for each hidden unit.

IV. EXPERIMENTAL CONFIGURATION

In this section, the experimental configuration, including datasets, evaluation methods, comparative approaches, and experimental setting is introduced.

A. DATA COLLECTIONS

We took two publicly available large-scale check-in datasets from real-world LBSNs, Foursquare, and Gowalla to conduct our experiment.

Foursquare data [35] included long-term (from 12 April 2012 to 16 February 2013) check-ins in New York and Tokyo. New York and Tokyo are the most populous metropolitan areas in the world, and the most popular check-in cities in USA and Asia respectively in Foursquare. Therefore, it is valuable and representative for the study of human mobility. Taking into account the urban compositions, the cultural differences of the two cities, and user check-in behaviors that exhibit different patterns, we conducted our experiment on the two cities using their datasets (i.e., Foursquare-NYC and Foursquare-TKY).

Gowalla data [36] consisted of check-ins in California and Nevada between February 2009 and October 2010. Since there is no significant cultural difference between these two adjacent areas, we conducted our experiment on the same dataset (i.e., Gowalla) without distinguishing cities.

We picked up 5-month check-in data in two datasets to conduct our experiment. Each check-in is a three-tuple $<$ user, venue, time $>$ and each venue is associated with the latitude and longitude. For each dataset, we removed users who had check-ins fewer than 4 locations each month, and locations which had been visited by fewer than 10 users. After preprocessing, the Foursquare (NYC) dataset contains 147,938 check-ins collected from 1083 users at 5135 locations, Foursquare (TKY) datasets contained 447,570 check-ins collected from 2293 users at 7873 locations, and Gowalla dataset contained 762,636 check-ins collected from 3374 users at 7208 locations. Basic statistics of the datasets are summarized in Table 2.

TABLE 2. Statistics of Two Datasets.

Datasets	City	#Users	#Venues	#Check-ins	Avg. #Check-ins
Foursquare	NYC	1083	5135	147,938	137
	TKY	2293	7873	447,570	95
Gowalla	--	3374	7208	762,636	226

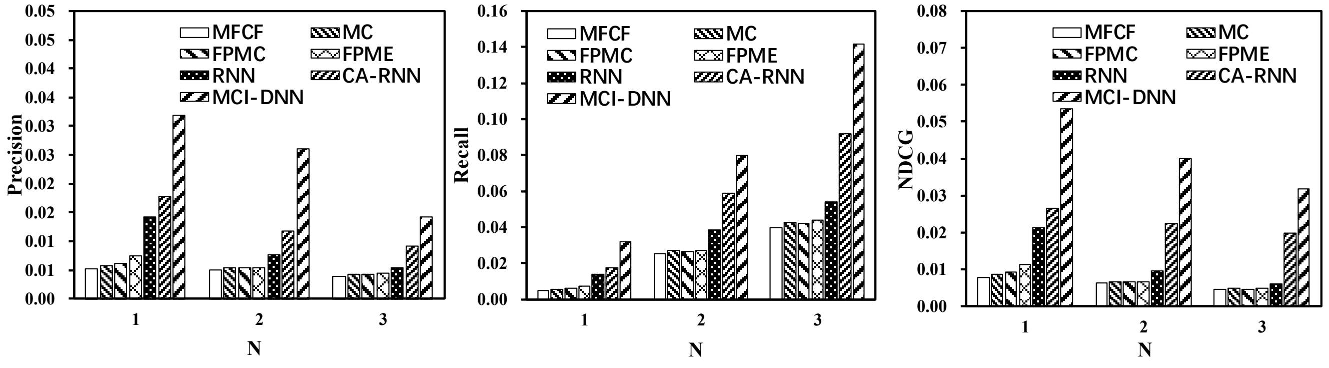
B. EVALUATION METHODS

As there is no explicit rating for test dataset, we evaluated our model based on the ranking list of the predicted locations. We presented each user with a certain number (N) of locations sorted by the predicted score using equation (8). We computed the precision, recall, F1-score and NDCG based on those locations which were visited by the user.

Precision and recall [8]. Given a top-N predicted location list $S_{u,t}^{N,pre}$ sorted in descending order of the prediction values, precision and recall are defined as:

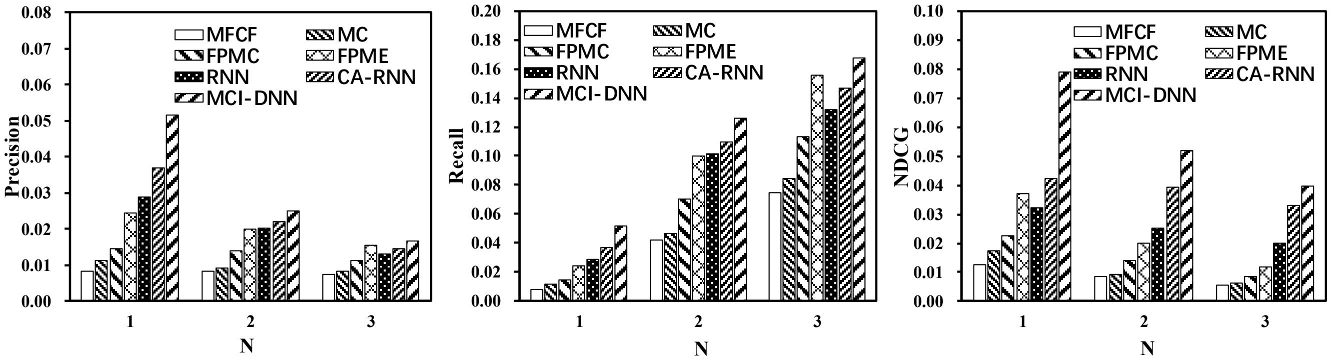
$$\text{Precision@N} = \frac{1}{|U|}\sum_{u=1}^{|U|}\sum_{t=1}^{L_{test}} \frac{S_{u,t}^{N,pre} \cap S_{u,t}^{visited}}{N} \tag{11}$$

$$\text{Recall@N} = \frac{1}{|U|}\sum_{u=1}^{|U|}\sum_{t=1}^{L_{test}} \frac{S_{u,t}^{N,pre} \cap S_{u,t}^{visited}}{S_{u,t}^{visited}} \tag{12}$$



(a) Precision (b) Recall (c) NDCG

Fig.2. Precision, Recall and NDCG of MCI-DNN and other approaches (MFCF, MC, FPMC, PRME, RNN, and CA-RNN) with the increasing N from 1 to 10 (with the interval of 5) using the Foursquare (NYC) dataset.



(a) Precision (b) Recall (c) NDCG

Fig.3. Precision, Recall and NDCG of MCI-DNN and other approaches (MFCF, MC, FPMC, PRME, RNN, and CA-RNN) with the increasing N from 1 to 10 (with the interval of 5) using the Foursquare (TKY) dataset.

where $S_{u,t}^{visited}$ are the locations a user has visited in the test data, L$_{\text{test}}$ is the length of the test sequence of each user. Note that the precision and recall are computed by averaging all the precision and recall values of all the users respectively.

F1-score [8]. An F1-score combines precision and recall. It is the harmonic mean of precision and recall.

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad (13)$$

NDCG@N is defined as: NDCG $= DCG@N/IDCG@N$, where DCG@N $= \sum_{i=1}^{N}(2^{rel_i} - 1)/log_{2(\text{i+1})}$. IDCG@N is equal to the DCG@N on condition that the recommended locations are ideally ranked, and rel_i refers to the graded relevance of the result ranked at the position i [37].

C. COMPARATIVE APPROACHES

We compared our MCI-DNN with the following six baseline approaches which representing the state-of-the-art location recommendation techniques:

Matrix Factorization based CF [38]: MFCF is the conventional collaborative filtering with matrix factorization, which factorizes the user-item preference matrix with BPR.

Markov Chains based model [10]: MC is a commonly used sequence model for sequence prediction, which computes the transition probability by a counting method.

Factorizing Personalized Markov chains model [9], [39]: FPMC is the state-of-art method that extends conventional MC methods and factorizes personalized probability transition matrices of users.

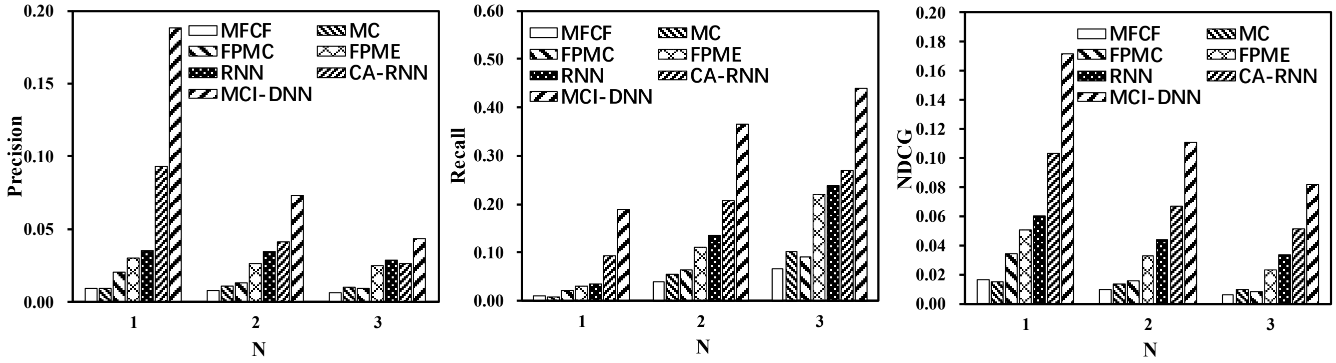
Personalized Ranking Metric Embedding [6]: PRME integrates sequence information, individual preference, and geographical influence to improve the recommendation performance.

Recurrent Neural Network [31]: RNN is the state-of-the-art method for sequence prediction, which has been successfully applied in natural language processing, click prediction and sequence recommendation task.

Context-Aware RNN [18]: CA-RNN is an extension of RNN for the sequence recommendation using adaptive context-specific matrices.

D. EXPERIMENTAL SETTING

According to the contextual information in the two datasets, similar to [18], [19], we extracted three kinds of input contexts: seven days in a week, twenty-four hours in a day, and time intervals between adjacent behaviors. For time interval, discretization was completed in one-day time bins. For those whose time intervals were larger than 30 days, they were treated as one time bin to avoid data sparsity. Note that our model is a generic and flexible model that can be extended to easily incorporate other input contexts that are not limited to



(a) Precision (b) Recall (c) NDCG

Fig.4. Precision, Recall and NDCG of MCI-DNN and other approaches (MFCF, MC, FPMC, PRME, RNN, and CA-RNN) with the increasing N from 1 to 10 (with the interval of 5) using the Gowalla dataset.

the above three input contexts. Moreover, we converted the location that a user visited into an *id* as input of MCI-DNN.

In our experiment, for each behavioral historical sequence in the two datasets, 80% check-ins of each user were selected for training data and remaining 20% for the test data. The learning rate of the proposed model was initialized as 0.1 and decay dynamically. Regularization parameter in the proposed model is set to 0.001 according to experimental results. Moreover, to avoid gradient explosion, the gradient range was limited to [-5, 5] using the clip gradient technique.

V. RESULTS AND DISCUSSION

A. PERFORMANCE COMPARISON

The results of our proposed MCI-DNN and other comparative approaches with well-tuned parameters are compared and reported by using the Foursquare and Gowalla datasets. It should be noted that only the performance when N is set to 1, 5, and 10 is shown because a greater value of N is usually ignored by users.

Figs. 2 and 3 show the precision, recall and NDCG achieved on the Foursquare datasets of NYC and TKY, and Table 3 shows the F1-score values. Results show that (a) our proposed MCI-DNN significantly outperforms all of the baseline methods. From Table 3, compared with CA-RNN, the relative improvements, in terms of F1-score@10, are more than 54.22% and 14.46% of NYC and TKY respectively; (b) MC-based approach significantly outperforms MFCF, which demonstrates the important influence of check-in sequences on human decision process [40]; (c) The performance of RNN significantly outperforms the MC-based model. This is because MC based approaches only model the transition probability between the current location and the latest visited location, while the influence of other previously visited locations was ignored [32]. Although a higher-order MC can capture the influence of other previously visited locations, it is subjected to huge prediction state space and computational complexity [1]. Compared to MC, RNN can capture relatively long sequence dependency. Thus, such a significant improvement of RNN is reasonable; (d) By taking the input and transition contexts into RNN, CA-RNN proposed by Liu et al. [18] performs better than RNN, indicating the importance of input context on the prediction of next locations.

TABLE 3. F1-score in the dataset from Foursquare.

City	@N	MFCF	MC	FPMC	PRME	RNN	CA-RNN	MCI-DNN
NYC	1	0.0051	0.0058	0.0061	0.0075	0.0142	0.0177	0.0319
	5	0.0085	0.0089	0.0089	0.0089	0.0128	0.0197	0.0393
	10	0.0072	0.0077	0.0076	0.0080	0.0098	0.0167	0.0257
TKY	1	0.0082	0.0114	0.0147	0.0243	0.0289	0.0369	0.0517
	5	0.0139	0.0154	0.0235	0.0334	0.0339	0.0366	0.0420
	10	0.0136	0.0153	0.0206	0.0283	0.0240	0.0266	0.0305

TABLE 4. F1-score in dataset from Gowalla

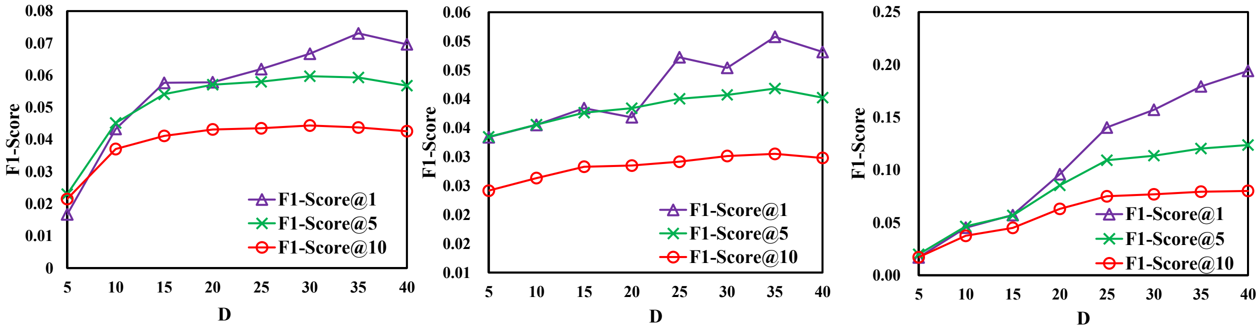
@N	MFCF	MC	FPMC	PRME	RNN	CA-NN	MCI-DNN
1	0.0096	0.0091	0.0204	0.0303	0.0357	0.0934	0.1882
5	0.0129	0.0181	0.0215	0.0430	0.0550	0.0692	0.1221
10	0.0119	0.0184	0.0166	0.0444	0.0516	0.0488	0.0799

Fig. 4 shows the precision, recall and NDCG achieved on the Gowalla dataset, and the corresponding F1-score values are shown in Table 4. We observe that the proposed MCI-DNN performed consistently better than all the baseline approaches. From Table 4, we can observe about 63.62% improvements in terms of F1-score@10 for MCI-DNN over CA-RNN. In addition, we observe that our model performed better on Gowalla than Foursquare in precision, recall, and F1-score and NDCG. The reason lies in the fact that each user's check-in data size in Gowalla is larger than Foursquare. As shown in Table 2, the average check-ins per user in Gowalla is about 65% and 15% larger than Foursquare (NYC) and Foursquare (TKY) dataset, which enable the model to capture users' preferences more accurately. Therefore, it is reasonable for the better performance of MCI-DNN on Gowalla dataset than Foursquare dataset.

The improvements in precision, recall, F1-score, and NDCG for the proposed MCI-DNN can be ascribed to the following reasons. Firstly, an MCI-DNN model with multiple elements was utilized in each hidden layer to capture the influence of sequence contexts. This is different from conventional RNN where only one element was considered. Visiting behaviors of a user are usually related to a series of related activities in a short time, making that the previous check-in behaviors have close connections to current and

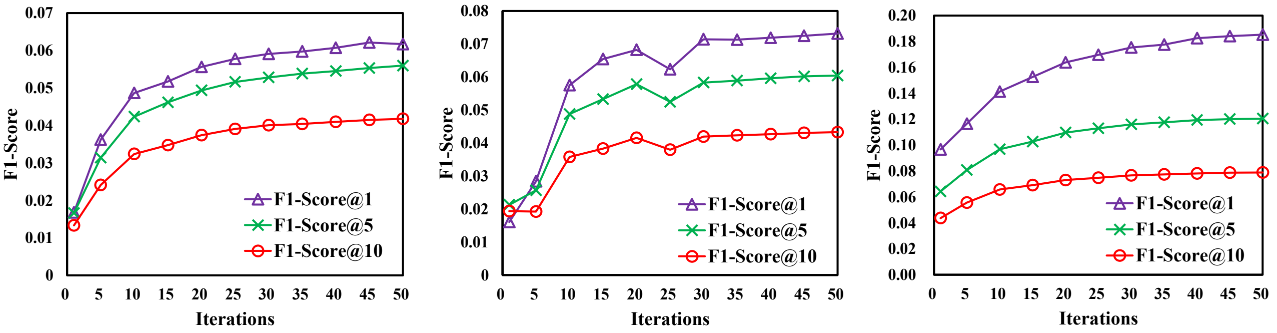
Table 5. Performance Comprise of MCI-DNN Variants using Datasets from Foursquare and Gowalla.

Datasets	Methods	Recall			Precision			NDCG		
		1	5	10	1	5	10	1	5	10
Foursquare (NYC)	MCI-DNN*	0.0200	0.0670	0.0999	0.0200	0.0134	0.0100	0.0393	0.0333	0.0306
	MCI-DNN	0.0319	0.0802	0.1414	0.0319	0.0260	0.0141	0.0536	0.0401	0.0319
Foursquare (TKY)	MCI-DNN*	0.0384	0.1211	0.1666	0.0384	0.0242	0.0167	0.0520	0.0494	0.0429
	MCI-DNN	0.0517	0.1260	0.1676	0.0517	0.0252	0.0168	0.0788	0.0519	0.0395
Gowalla	MCI-DNN*	0.1087	0.2918	0.4006	0.1087	0.0584	0.0401	0.1233	0.0809	0.0682
	MCI-DNN	0.1883	0.3665	0.4396	0.1883	0.0733	0.0440	0.1719	0.1106	0.0822



(a) Foursquare(NYC) (b) Foursquare(TKY) (c) Gowalla

Fig.5. Impact of Dimension of Hidden Layer on the performance of MCI-DNN at different predicted location number using the Datasets from (a) Foursquare(NYC), (b) Foursquare (TKY) and (c) Gowalla.



(a) Foursquare(NYC) (b) Foursquare(TKY) (c) Gowalla

Fig.6. Convergence curves of F1-Score using MCI-DNN at different predicted location numbers using the dataset (a) Foursquare (NYC), (b) Foursquare (TKY), (c) Gowalla.

future decision [9], [41], [6]. Therefore, the performance of predicting next locations is significantly improved by considering sequence contexts. We will report the experimental results in Section 5.2. Secondly, the interaction of different kinds of input context was considered in our model, and the embedding representation technology was adopted to avoid the data sparsity of input contexts. Although CA-RNN models different kinds of input contexts using context-specific projection matrices, it fails to capture the interaction of different kinds of input contexts. Thirdly, user preferences were considered to enhance the performance of next location prediction. In Table 5, MCI-DNN* is the variant of MCI-DNN in which user preferences were not taken into account. We observe that we achieved about 16.43% improvement on average in term of recall@5 and 41.2% improvement on average in term of precision@5.

B. *PARAMETER SENSITIVITY ANALYSIS*

TABLE 6. Impact of Window Width on the Performance of MCI-DNN Using Foursquare (NYC).

N Metric	1	3	5	7	9	11
Precision	0.0051	0.0091	0.0129	0.0197	0.0208	0.0192
Recall	0.0513	0.0911	0.1288	0.1973	0.2079	0.183
F1-Score	0.0093	0.0165	0.0234	0.0359	0.0378	0.0350

TABLE 7. Impact of Window Width on the Performance of MCI-DNN Using Foursquare (TKY).

N Metric	1	3	5	7	9	11
Precision	0.0108	0.0121	0.0134	0.0142	0.0146	0.0133
Recall	0.1083	0.1212	0.1335	0.1418	0.1461	0.1325
F1-Score	0.0197	0.0220	0.0243	0.0258	0.0265	0.0249

In this experiment, we investigated the impact of the window width and the dimension of the hidden layer on the performance of the proposed MCI-DNN using the Foursquare and Gowalla datasets.

Tables 6 and 7 depict the influences of window width n on the prediction accuracy for the Foursquare dataset. Note that results regarding the Gowalla dataset are similar to those regarding the Foursquare dataset, and thus are not presented here. The parameter n being set to 1 only means that the current element was considered as input to RNN. We observe that with the increase of window width in two datasets, the performance improves significantly. The smaller window width results in a worse performance. In this case, only a few previous behaviors were considered. In contrast, the larger window width results in better performance. This is attributable to the fact that most of the successive check-ins occurred within a short period, such as two hours. Hence, the result is consistent with the finding in [9], which reported that almost 40% and 48% successive check-ins occurred within two hours in Foursquare and Gowalla respectively.

By setting window width n as 9, we further studied the impact of D on the prediction accuracy using the Foursquare and Gowalla datasets. Fig. 5 shows the result of F1-score values with the variation of D from 5 to 40 (with the interval of 5). The result shows that with the increase of D in three datasets, the F1-score values become higher, indicating better performance. However, when D was higher than 35, lower F1-score values were obtained in Foursquare (NYC) and Foursquare (TKY) datasets illustrated in Figs. 5(a) and 5(b). This phenomenon implies that overfitting may occur when D is very large. Moreover, for the Gowalla dataset, F1-score changed smoothly when D was larger than 35, but the trend of F1-score was not obvious. Then, according to the curves, the best dimension size can be set to 30, 35, and 40 for Foursquare (NYC), Foursquare (TKY), and Gowalla, respectively.

C. EFFICIENCY ANALYSIS

We further investigated the computational time and the convergence of the learning progress of the proposed method. Fig.6 illustrates the convergence curves of F1-score that were obtained by using the proposed method on three datasets. Results show that MCI-DNN converged in a relatively small number of iterations. For the Foursquare (NYC) dataset, the learning process converged in about 30 iterations, while the learning process converged in about 35 iterations on the Foursquare (TKY) and 45 on the Gowalla dataset. This is because the average check-in number per user for the Foursquare (NYC) is smaller than the Foursquare (TKY) and the Gowalla dataset.

TABLE 8. Comparing of Computation Time.

Datasets / Methods	Foursquare (NYC)	Foursquare (TYK)	Gowalla
RNN	11s	27s	10s
CA-RNN	23s	65s	57s
MCI-DNN	39s	126s	87s

The results of computational efficiency in each iteration are shown in Table 8. The computation time was measured in seconds. We observe that all these methods had relatively short training time. Although the computation time of MCI-DNN is longer than RNN and CA-RNN, the computation time of MCI-DNN with a significant performance improvement is still acceptable.

VI. CONCLUSION

We have presented an integrated analysis of the joint effect of multiple factors, i.e., sequence context, input contexts, and user preferences, on the process of a user's decision to the next location. We have also developed an effective Multi-Context Integrated Deep Neural Network Model (MCI-DNN) to improve the accuracy of next location prediction. The prediction results by two datasets from Foursquare and Gowalla demonstrate the significant joint influence of sequence context and the interaction of different kinds of input contexts on the user's decision to the next locations. The average improvement in term of F1-score@5 was about 57.12% for Foursquare and 76.4% for Gowalla. The model developed herein performed better than the state-of-the-art approaches in view of prediction accuracy and stability. The proposed method shows significant potential for next location predictions in several applications where sequence context and input context characteristics exist, such as a recommendation system, advertising delivery, traffic jams forecasting, urban planning and so on.

REFERENCES

[1] W. Wang, H. Yin, L. Chen, Y. Sun, S. Sadiq, and X. Zhou, “ST-SAGE: A Spatial-Temporal Sparse Additive Generative Model for Spatial Item Recommendation,” *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 3, pp. 1–25, Apr. 2017.

[2] H. Yin, W. Wang, H. Wang, L. Chen, and X. Zhou, “Spatial-Aware Hierarchical Collaborative Deep Learning for POI Recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 11, pp. 2537–2551, Nov. 2017.

[3] J. Wang, Q. Qi, S. Qing, and J. Liao, “Elastic Vehicular Resource Providing Based on Service Function-Group Resource Mapping of Smart Identify Network,” *IEEE Systems Journal*, pp. 1–12, 2017.

[4] H. Yin, X. Zhou, B. Cui, H. Wang, K. Zheng, and Q. V. H. Nguyen, “Adapting to User Interest Drift for POI Recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 10, pp. 2566–2581, Oct. 2016.

[5] J. Ye, Z. Zhu, and H. Cheng, “What’s your next move: User activity prediction in location-based social networks,” in *Proceedings of the 2013 SIAM International Conference on Data Mining*, 2013, pp. 171–179.

[6] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, “Personalized Ranking Metric Embedding for Next New POI Recommendation,” in *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, New York, NY, USA, 2015, pp. 2069–2075.

[7] T. H. N. Vu, K. H. Ryu, and N. Park, “A method for predicting future location of mobile user for location-based services system,” *Computers & Industrial Engineering*, vol. 57, no. 1, pp. 91–105, Aug. 2009.

[8] B. Liu, H. Xiong, S. Papadimitriou, Y. Fu, and Z. Yao, “A General Geographical Probabilistic Factor Model for Point of Interest Recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 5, pp. 1167–1179, May 2015.

[9] C. Cheng, H. Yang, M. R. Lyu, and I. King, “Where You Like to Go Next: Successive Point-of-Interest Recommendation,” in *Proceedings of the 23th International joint Conference on Artificial Intelligence*, Beijing, China, 2013, pp. 2605–2611.

[10] J.-D. Zhang and C.-Y. Chow, “Spatiotemporal Sequential Influence Modeling for Location Recommendations: A Gravity-based Approach,”

ACM Transactions on Intelligent Systems and Technology, vol. 7, no. 1, pp. 1–25, Oct. 2015.

[11] X. Liu, Y. Liu, and X. Li, “Exploring the Context of Locations for Personalized Location Recommendations,” in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, New York, NY, USA, 2016, pp. 1188–1194.

[12] S. Zhao, T. Zhao, I. King, and M. R. Lyu, “Geo-Teaser: Geo-Temporal Sequential Embedding Rank for Point-of-interest Recommendation,” in *Proceedings of the 26th International Conference on World Wide Web Companion*, Perth, AUS, 2017, pp. 153–162.

[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 30th AAAI Conference on Artificial Intelligence*, Phoenix, AZ, USA, 2016, pp. 194–200.

[14] C. Yang, M. Sun, W. X. Zhao, Z. Liu, and E. Y. Chang, “A Neural Network Approach to Joint Modeling Social Networks and Mobile Trajectories,” *ACM Transactions on Information Systems*, vol. 35, no. 4, p. 36, Aug. 2017.

[15] Q. Cui, S. Wu, Y. Huang, and L. Wang, “A Hierarchical Contextual Attention-based GRU Network for Sequential Recommendation,” *arXiv preprint arXiv:1711.05114*, 2017.

[16] N. Zhou, W. X. Zhao, X. Zhang, J. R. Wen, and S. Wang, “A General Multi-Context Embedding Model for Mining Human Trajectory Data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 8, pp. 1945–1958, Aug. 2016.

[17] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang, “Learning Graph-based POI Embedding for Location-based Recommendation,” in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, Indianapolis, IN, USA, 2016, pp. 15–24.

[18] Q. Liu, S. Wu, D. Wang, Z. Li, and L. Wang, “Context-aware Sequential Recommendation,” in *Proceedings of the 16th IEEE International Conference on Data Mining*, Barcelona, ESP, 2016, pp. 1053–1058.

[19] H. Jing and A. J. Smola, “Neural Survival Recommender,” in *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, NY, USA, 2017, pp. 515–524.

[20] M. Ye, P. Yin, W. C. Lee, and D. L. Lee, “Exploiting geographical influence for collaborative point-of-interest recommendation,” in *Proceedings of the 34th International ACM SIGIR conference on Research and development in Information Retrieval*, Beijing, China, 2011, pp. 325–334.

[21] M. Ye, D. Shou, W.-C. Lee, P. Yin, and K. Janowicz, “On the semantic annotation of places in location-based social networks,” in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, USA, 2011, pp. 520–528.

[22] X. Li, G. Cong, X.-L. Li, T.-A. N. Pham, and S. Krishnaswamy, “Rank-GeoFM: A Ranking based Geographical Factorization Method for Point of Interest Recommendation,” in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Santiago, Chile, 2015, pp. 433–442.

[23] Z. Yang, B. Wu, K. Zheng, X. Wang, and L. Lei, “A Survey of Collaborative Filtering-Based Recommender Systems for Mobile Internet Applications,” *IEEE Access*, vol. 4, pp. 3273–3287, 2016.

[24] 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*, 2010, pp. 211–222.

[25] X. Ren, M. Song, H. E, and J. Song, “Context-aware probabilistic matrix factorization modeling for point-of-interest recommendation,” *Neurocomputing*, vol. 241, no. C, pp. 38–55, Jun. 2017.

[26] S. Zhao, T. Zhao, H. Yang, M. R. Lyu, and I. King, “STELLAR: Spatial-Temporal Latent Ranking for Successive Point-of-Interest Recommendation,” in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, Phoenix, AZ, USA, 2016, pp. 315–322.

[27] H. Zhu, E. Chen, H. Xiong, K. Yu, H. Cao, and J. Tian, “Mining Mobile User Preferences for Personalized Context-Aware Recommendation,” *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 4, pp. 1–27, Dec. 2014.

[28] S. Feng, G. Cong, B. An, and Y. M. Chee, “POI2Vec: Geographical Latent Representation for Predicting Future Visitors,” in *Proceedings of the 31th AAAI Conference on Artificial Intelligence*, San Francisco, CA, USA, 2017, pp. 102–108.

[29] J. Wang *et al.*, “MindCamera: Interactive Sketch-Based Image Retrieval and Synthesis,” *IEEE Access*, vol. 6, pp. 3765–3773, 2018.

[30] H. Guo, “Generating Text with Deep Reinforcement Learning,” *Computer Science*, vol. 40, no. 4, pp. 1–5, Oct. 2015.

[31] S. Wu, W. Ren, C. Yu, G. Chen, D. Zhang, and J. Zhu, “Personal recommendation using deep recurrent neural networks in NetEase,” in *2016 IEEE 32nd International Conference on Data Engineering*, Helsinki, FIN, 2016, pp. 1218–1229.

[32] Q. Liu, S. Wu, and L. Wang, “Multi-behavioral Sequential Prediction with Recurrent Log-bilinear Model,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 6, pp. 1254–1267, Jan. 2017.

[33] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng, “Improving word representations via global context and multiple word prototypes,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers*, Jeju Island, Korea, 2012, pp. 873–882.

[34] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “BPR: Bayesian personalized ranking from implicit feedback,” in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, Montreal, Quebec, Canada, 2009, pp. 452–461.

[35] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, “Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 129–142, Jun. 2015.

[36] E. Cho, S. A. Myers, and J. Leskovec, “Friendship and mobility: user movement in location-based social networks,” in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2011, pp. 1082–1090.

[37] Y. Liu, T.-A. N. Pham, G. Cong, and Q. Yuan, “An experimental evaluation of point-of-interest recommendation in location-based social networks,” *Proceedings of the VLDB Endowment*, vol. 10, no. 10, pp. 1010–1021, 2017.

[38] A. Mnih and R. R. Salakhutdinov, “Probabilistic matrix factorization,” in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, Vancouver, British Columbia, Canada, 2007, pp. 1257–1264.

[39] 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*, Raleigh, North Carolina, USA, 2010, pp. 811–820.

[40] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, “Mining interesting locations and travel sequences from GPS trajectories,” in *Proceedings of the 18th international conference on World wide web*, 2009, pp. 791–800.

[41] H. Gao, J. Tang, X. Hu, and H. Liu, “Exploring temporal effects for location recommendation on location-based social networks,” in *Proceedings of the 7th ACM conference on Recommender system*, Hong Kong, China, 2013, pp. 93–100.