Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.055141

ARTICLE

Adaptive Successive POI Recommendation via Trajectory Sequences Processing and Long Short-Term Preference Learning

Yali Si[1,2], Feng Li[1,*], Shan Zhong[1,2], Chenghang Huo[3], Jing Chen[4] and Jinglian Liu[1,2]

[1]School of Computer Science and Engineering, Changshu Institute of Technology, Changshu, 215500, China

[2]Suzhou Industrial Intelligence and Big Data Laboratory, Suzhou, 215000, China

[3]School of Information Science and Engineering, Yanshan University, Qinhuangdao, 066004, China

[4]College of Electronic and Information Engineering, Guangdong Ocean University, Zhanjiang, 524088, China

*Corresponding Author: Feng Li. Email: ysu_lifeng@cslg.edu.cn

Received: 18 June 2024 Accepted: 27 August 2024 Published: 15 October 2024

ABSTRACT

Point-of-interest (POI) recommendations in location-based social networks (LBSNs) have developed rapidly by incorporating feature information and deep learning methods. However, most studies have failed to accurately reflect different users' preferences, in particular, the short-term preferences of inactive users. To better learn user preferences, in this study, we propose a long-short-term-preference-based adaptive successive POI recommendation (LSTP-ASR) method by combining trajectory sequence processing, long short-term preference learning, and spatiotemporal context. First, the check-in trajectory sequences are adaptively divided into recent and historical sequences according to a dynamic time window. Subsequently, an adaptive filling strategy is used to expand the recent check-in sequences of users with inactive check-in behavior using those of similar active users. We further propose an adaptive learning model to accurately extract long short-term preferences of users to establish an efficient successive POI recommendation system. A spatiotemporal-context-based recurrent neural network and temporal-context-based long short-term memory network are used to model the users' recent and historical check-in trajectory sequences, respectively. Extensive experiments on the Foursquare and Gowalla datasets reveal that the proposed method outperforms several other baseline methods in terms of three evaluation metrics. More specifically, LSTP-ASR outperforms the previously best baseline method (RTPM) with a 17.15% and 20.62% average improvement on the Foursquare and Gowalla datasets in terms of the F_β metric, respectively.

KEYWORDS

Location-based social networks; adaptive successive point-of-interest recommendation; long short-term preference; trajectory sequences

1 Introduction

The popularity of portable intelligent devices has greatly promoted the development of location-based social networks (LBSNs) [1], such as X (formerly Twitter), WeChat, Foursquare, Gowalla and Dianping. Users often prefer to socialize and share location-tagged life experiences by making friends online, checking in locations, or commenting on posts on mobile social networks. Particularly, according to statistics, Facebook currently has 2.09 billion daily visitors and 3.05 billion monthly active users. Within these massive records of social relations, comments, and check-in information, there exists a wealth of user features, such as group preferences, geography, check-in periodicity, and user movement trajectory sequences [2]. Therefore, one of the major challenges for LBSNs is to extract the implicit features from the massive datasets and accurately apply them for point-of-interest (POI) recommendations to effectively reduce the selection confusion caused by location overload.

In POI recommendation studies, the main task involves recommending personalized and precise locations for users by mining preferences from LBSN historical check-in datasets [3]. Importantly, the recommended locations must be those that the users have not previously visited. Existing methods have modeled user check-in behavior and preferences using collaborative filtering, matrix factorization [4], probabilistic models and deep learning (DL) methods [5,6]. The efficiency and accuracy of POI recommendation are improved by integrating time, geographical factors, social relationships, POI popularity, neighborhood characteristics, and several other factors [7,8].

However, in real life, user behavior for visiting locations generally exhibits the characteristics of continuity, regional restrictions, and time sensitivity. For example, IT employees may go to nearby restaurants at noon on weekdays, coffee shops in the afternoon, and then the gym in the evening. In this scenario, after visiting restaurants and cafes, the next recommended POI would reasonably be a gym. Therefore, the current context information (i.e., time, geographical location, and region) and historical check-in track sequences of the LBSN users must be deeply explored to provide valuable and practical successive POI recommendations (Fig. 1).

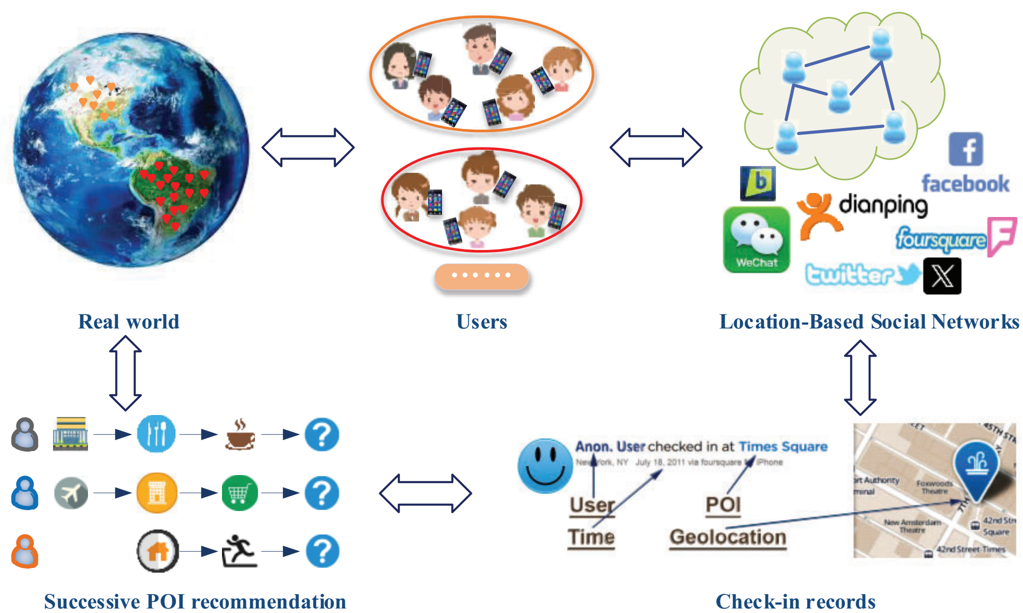


Figure 1: Successive POI recommendation in LBSNs

Compared to traditional POI recommendation methods, which utilize the entire user check-in records, successive POI recommendation focuses more on the modeling of time relationships and check-in trajectory sequences [9]. This methodology aims to utilize the sparser continuous check-in sequences to provide POI recommendations not only based on user interests, but also on contextual conditions, making it a more challenging task. Existing successive POI recommendation approaches employ the Markov chain or DL to model the continuous sign-in behavior of users, considering the influence of recent and long-term check-in records, as well as the spatiotemporal context of two adjacent check-in locations [10]. These methods have effectively promoted the research and development of POI recommendations. However, with the diversification of user mobility behaviors, these schemes remain unable to provide high-performance recommendations for real-life applications due to the following reasons:

- Inactive check-in behavior: In mobile social networks, some users can have lower activity, fewer check-ins, and longer time and distance intervals between adjacent POIs. These inactive users will have short check-in track sequences and poor continuity. With a lack of sufficient recent check-in records, it can be quite difficult to learn short-term interests and capture user preferences accurately to provide good recommendations.
- User preference diversity within continuous check-in sequences: Most existing POI recommendation methods consider the complete user check-in record to generate a continuous sequence, without distinguishing between historical and recent check-in information. Few methods that do consider short-term interests depend on the most recently checked-in location, ignoring the impact of other recent continuous check-in locations on POI recommendation.
- Lack of adaptive learning mechanisms: Existing methods use a single DL model for the check-in trajectories of all LBSN users instead of an adaptive learning mechanism for short- and long-term preferences for different trajectory sequences, resulting in the inability to precisely extract different types of interests for multiple users.

To solve the abovementioned problems, we propose a **l**ong-**s**hort-**t**erm-**p**reference-based **a**daptive **s**uccessive POI **r**ecommendation (LSTP-ASR) method with appropriate sequence processing and spatiotemporal context. To reflect user interest characteristics over different periods, each user check-in trajectory is divided into a recent and a historical trajectory sequence according to a dynamic time window. To achieve adaptive learning, we applied recurrent neural network (RNN) and long short-term memory (LSTM) models on temporal and spatial factors to extract the short- and long-term user interests, respectively. The major contributions of our work can be summarized as follows:

- We design an adaptive sequence processing strategy that utilizes time windows and sequence filling. The dynamic time window divides the user check-in track sequences into recent and historical sequences to better reflect user interest diversity. The sequence filling method expands the recent check-in track sequences of inactive users using those of similar active users to effectively solve the problems of limited recent check-in records and cold start.
- We propose a flexible adaptive model to learn long short-term interests that can accurately obtain different user preferences. The RNN with spatiotemporal context and the LSTM network with temporal context are used to model the short-and long-term interests, respectively. Furthermore, an adaptive successive POI recommendation algorithm, i.e., LSTP-ASR, is proposed.
- Experiments on two datasets reveal that the proposed LSTP-ASR algorithm outperforms other baseline algorithms; the results also indicate the effectiveness of key LSTP-ASR components.

The remainder of this paper is organized as follows: In Section 2, we discuss the related works on successive POI recommendations; in Section 3, we describe the components of the proposed method in detail; In Sections 4 and 5, we analyze the experimental results and discuss the practical applicability of the proposed model; and finally, in Section 6, we address our primary conclusions and future work directions.

2 Literature Review

To comprehensively understand successive POI recommendation systems, in this section, we review previous studies and divide them into the following Markov-chain- and DL-based methods.

2.1 Markov-Model-Based Methods

In the Markov-model-based methods, the first-order Markov chain is used to model the transfer matrix between continuous user check-in locations, and subsequently, the third-order tensor model and matrix factorization are integrated to realize successive POI recommendations. Cheng et al. [11] first proposed the matrix factorization model and adopted a personalized first-order Markov chain to extract the continuous check-in behavior of users. By including local region restrictions, this method improves position correlations and computational efficiency. Feng et al. [12] introduced an extended ranking metric embedding model that integrates three influencing factors and avoids the data sparsity problem caused by matrix factorization. He et al. [13] designed a Bayesian personalized ranking model using a potential pattern-based Markov chain and a third-order tensor to better extract and optimize continuous check-in behaviors. Zhao et al. [14] built STELLAR on a spatiotemporal ranked pairwise tensor decomposition frame and conducted detailed modeling on the interactions between users, locations, and time. Chen et al. [15] designed and applied a spatiotemporal probabilistic location prediction model that integrated multiple dynamic mobility features in the Naïve Bayes algorithm, on geotagged social media data. Apart from these studies, there also exist two-step methods with POI categories, grid-based regional models, and temporal metric embedding methods with non-symmetrical projection [16] for POI recommendation.

As previously mentioned, successive POI recommendation requires a comprehensive consideration of the users' current locations, as well as their previous trajectory sequences and preferences. However, Markov models have a strong assumption regarding successive behavior that the next moment state is only related to the current state. As a result, these models cannot record prior checked-in locations, resulting in the loss of historical information and long-term preferences.

2.2 DL-Based Methods

DL, as one of the most advanced subsets of artificial intelligence methods, has been successfully applied to model sequence data in different fields. For POI recommendation, DL has been used to model users' successive check-in behavior [17], showing significant advantages over Markov models. In RNNs, transition matrices of abundant features have been utilized to better learn user preferences, such as temporal context, distance information [18], mobile trajectories, and relationships [19]. Zhang et al. [20] established NEXT, a unified neural network (NN) framework, to model the user's hidden intent using sequential influence, temporal factor, geographical context, and metadata. Lu et al. [21] proposed a two-step model for successive POI recommendation: (i) initially, they split an area into grids to estimate the regional influence and applied edge-weighted personalized PageRank in the location transition model; and then, (ii) the model fused the successive transition factor, regional factor, and user interests into a uniform framework using word embeddings and RNNs.

By integrating the attention mechanism into successive POI recommendation methods, the attention coefficient of different variables can be learned to explain their correlations. LSTM models combining spatiotemporal factors have been proposed to better extract long- and short-term interests between successive check-in records [22–25]. Li et al. [26] constructed a multi-modal heterogeneous graph by combining five types of check-in information and applied an attentional RNN to make POI recommendations. Wang et al. [27] considered both real-time requirements and user interests. For real-time modeling, they used multiple contexts, including location-location transition distance and time, location category, and absolute time. Further, they adopted an attention-based RNN model to automatically learn these contexts and user interests. Liu et al. [28] constructed a real-time interest mining model using an LSTM network with time restrictions: they mined weekly periodic trends to indicate long-term behavior, represented the public interest of each time slot as a trainable time transition vector, and integrated it into the current interest model for the short term preferences. Liu et al. [29] provided group recommendations based on a bipartite graph neural network (GNN) with edge learning enhancement and model similar users' POI interaction interests. They further proposed a session-based GNN to extract similar users' location transfer interests.

Most DL-based POI recommendation approaches utilize one model for all LBSN users, neglecting the diversity of preferences and self-adaption of learning models. Additionally, there remains a lack of effective solutions for recent check-in sequence extraction and short-term preference learning for inactive users with limited check-in records. The proposed LSTP-ASR model comprehensively considers and resolves the abovementioned challenges to realize effective POI recommendations.

3 Methodology

3.1 Notations and Definitions

For ease of presentation, all notations used in this paper and their descriptions are summarized in Table 1. We further define the following necessary definitions:

Table 1: Notations and their descriptions

Notations	Descriptions
U_{all}	Check-in dataset of a LBSN
L	Location set of a LBSN
T	Time and date set
U	User set of a LBSN
T_U	Check-in trajectory sequences of all users
tw_u	Time window of a user u
C_u	Check-in trajectory of a user u
C_u^{tw}	Recent trajectory of u
C_u^{hw}	History trajectory sequence of u
p_i	Users' short-term preferences
p_k	Users' long-term interests
U_{in}	Set of inactive users
U_{ac}	Set of active users
$S_{u_{in},u_{ac}}$	Similarity of inactive user u_{in} and active user u_{ac}
G	Set of similar users
$C_{u_{in}}^{tw}$	Recent trajectory sequence of inactive user u_{in}
L_{lc}	Set of candidate POIs
$S_{u,lc}$	Predicted probability values of u visiting new candidate locations

Definition 1. User check-in trajectory sequence: Let $C_u = \{c^u_{t_1}, \cdots, c^u_{t_k}, \cdots, c^u_{t_i}, \cdots, c^u_{t_c}\}$ (where $t_i \in T$) denote the entire check-in trajectory sequence set of a user $u \in U$, which includes all consecutive check-in records in adjacent time slots. The set of complete check-in trajectory sequences of all users is denoted by $T_U = \{C_{u_1}, C_{u_2}, \cdots, C_{u_{|U|}}\}$, where $|U|$ is the total number of LBSN users.

Definition 2. User recent check-in trajectory sequence: Let $C^{tw}_u = \left\{c^u_{tw_u}, \cdots, c^u_{t_i}, \cdots, c^u_{t_c}\right\}$ (where $C^{tw}_u \subseteq C_u$) denote the recent trajectory sequence of u in time window tw_u. The check-in time t_i meets the condition $tw_u \le t_i \le t_c$, where t_c is the current time.

Definition 3. User historical check-in trajectory sequence: Let $C^{hw}_u = \{c^u_{t_1}, \cdots, c^u_{t_k}, \cdots, c^u_{tw_u}\}$ (where $C^{hw}_u \subseteq C_u$) denote the historical check-in trajectory sequence set of u, i.e., the early check-in records. The check-in time t_k meets the condition $t_1 \le t_k \le tw_u$ and $C_u = C^{hw}_u \cup C^{tw}_u$, where t_1 is the first check-in time.

3.2 Adaptive Successive POI Recommendation Framework

Considering user behavior diversity, it is essential to design adaptive trajectory sequence division and long short-term preference learning for successive POI recommendations. The proposed LSTP-ASR model (Fig. 2), after data preprocessing, includes the following main stages:

(1) Adaptive check-in sequence processing: Trajectory sequences for different types of users are dynamically processed and divided into recent (C^{tw}_u) and historical (C^{hw}_u) check-in track sequences according to the size of the time window, tw_u. Short recent check-in sequences of inactive users were adaptively filled using the recent records of similar active users, thus, effectively extracting short-term preferences.

(2) Adaptive short- and long-term preference learning: User's short- and long-term features are extracted according to the recent and historical check-in sequences, respectively. The feature representation of short-term preference, $p_i \in \mathbb{R}^d$, is a d-dimensional feature vector learned from a recent check-in sequence C^{tw}_u. To model short-term preferences, we considered the influence of spatiotemporal context and integrated it into the RNN model. The feature representation of long-term and stable preferences, $p_k \in \mathbb{R}^d$, is obtained from the LSTM model applied to the temporal contexts of the historical check-in sequence C^{hw}_u.

(3) Successive POI recommendation: The candidate POIs are first selected according to the distance feature. Then the predicted probability values of candidate POIs are calculated using the inner product of short and long representations. Finally, the top-n POIs are recommended to the users.

Unlike traditional successive POI recommendation methods, LSTP-ASR can realize adaptive sequence processing and adaptive long short-term preferences modeling. Importantly, short- and long-term preference features vary dynamically with the increase in user check-in sequences, which can allow this model to fully reflect the current interest features of users. The abovementioned processes are discussed below in detail.

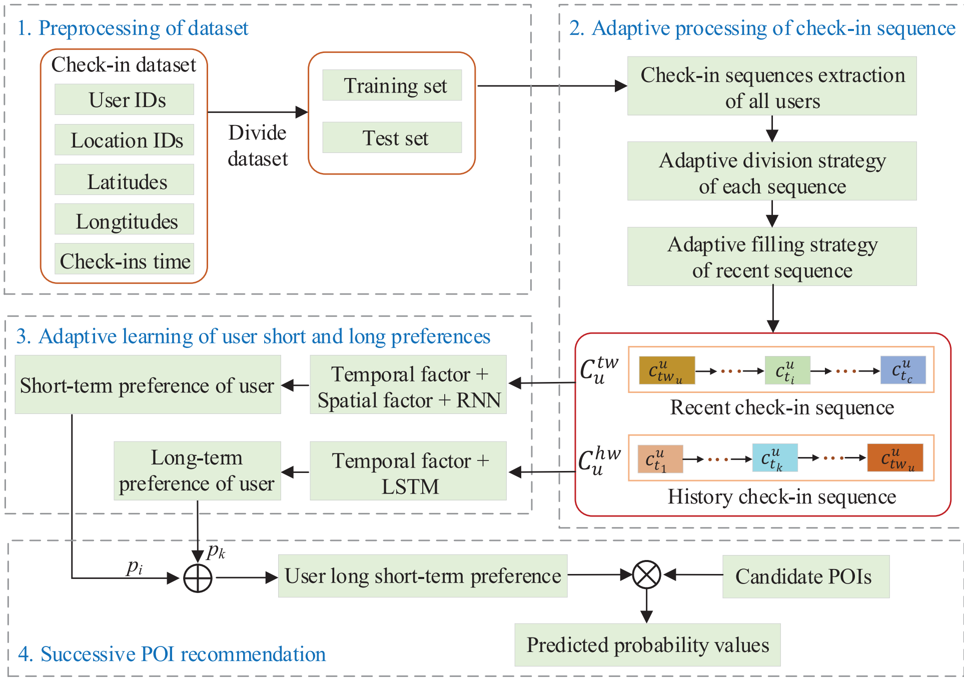


Figure 2: Overview of the proposed LSTP-ASR model

3.3 Adaptive Check-in Sequence Processing

3.3.1 Sequence Extraction and Adaptive Division

As the user check-in trajectory and frequency vary, we used a dynamic time window for adaptive division to better reflect the personalized characteristics of users. For each user, the trajectory was divided by the longest adjacent check-in interval. The steps for the sequence extraction and adaptive division strategy are detailed as follows:

(1) Sequence extraction: Let $c_i^u =< u, l_i, Lon_i, Lat_i, time_i, date_i >$ denote a check-in record of a user, where $u \in U$, l_i, Lon_i, Lat_i, $time_i$, and $date_i$ represent the user ID, POI ID, longitude, latitude, check-in time, and date, respectively. For a user u, we sort the date, followed by the time in ascending order to obtain a check-in trajectory sequence set, $C_u = \{c_{t_1}^u, \cdots, c_{t_k}^u, \cdots, c_{t_i}^u, \cdots, c_{t_c}^u\}$.

(2) Dynamic time window calculation: For each user, the time difference, Δt_i, of all pairs of adjacent records is calculated as

$$\Delta t_i = t_i - t_{i-1} = (date_i - date_{i-1}) \times 24 + (time_i - time_{i-1}) \tag{1}$$

and the time intervals are used as the basis for selecting the time window. The instance with maximum time difference is selected as the segmentation node for dividing the trajectory sequence. Thus, the dynamic time window, tw_u, of a user can be defined as

$$tw_u = t(\max(\Delta t_2, \Delta t_3, \cdots, \Delta t_i)) \tag{2}$$

(3) Adaptive division strategy: We then use tw_u to divide C_u. The trajectory sequence from the earliest check-in to the instance tw_u is defined as the historical sequence, $C_u^{hw} = \{c_{t_1}^u, \cdots, c_{t_k}^u, \cdots, c_{tw_u}^u\}$. The

check-in trajectory sequence post tw_u is defined as the recent sequence, $C_u^{tw} = \left\{c_{tw_u}^u, \cdots, c_{t_i}^u, \cdots, c_{t_c}^u\right\}$. Here, $C_u^{tw} \subseteq C_u$, $C_u^{hw} \subseteq C_u$, and $C_u = C_u^{hw} \cup C_u^{tw}$.

3.3.2 Adaptive Filling of Recent Sequence

The short-term interests of users are primarily obtained from recent check-in trajectory sequences. However, low-activity users with fewer check-in records hinder the accurate capture of short-term preferences. In real life, several users can have similar traveling behavior and activity patterns (e.g., IT employees, university teachers, and college students). These users with common interests often engage in similar activities in the same regions. Therefore, the short-term preferences of inactive users can be accurately inferred using those of similar active users.

To better learn short-term interests and solve the cold start problem of inactive users, we employ an adaptive filling strategy of recent sequence. Then a DL model is adopted to learn users' recent check-in sequences to obtain short-term interests. The steps for adaptive sequence filling are as follows:

(1) Let C_u^{tw} be the recent check-in trajectory sequence of user u and δ_{min} be the minimum number of check-in records required by the recent sequence. If $|C_u^{tw}| < (>)\, \delta_{min}$, the sequence has very few (sufficient) check-in records, and therefore, u is assigned to the inactive (active) user set, U_{in} (U_{ac}).

(2) For each inactive user $u_{in} \in U_{in}$, we calculate the similarity values with active users $u_{ac} \in U_{ac}$. We can generally categorize u_{in} and u_{ac} as similar users if they have visited the same locations. We calculate the similarity degree between u_{in} and u_{ac} using the classic cosine similarity, $S_{u_{in},u_{ac}}$, defined as

$$S_{u_{in},u_{ac}} = \frac{\sum_l \left(r_{u_{in},l} \cdot r_{u_{ac},l}\right)}{\sqrt{\sum_l \left(r_{u_{in},l}\right)^2} \cdot \sqrt{\sum_l \left(r_{u_{ac},l}\right)^2}} \tag{3}$$

where $r_{u_{in},l}$ is a binary value (1 if u_{in} visited and checked-in at location l and 0 otherwise). The similarity between two users increases with the number of same locations they have both visited. For an inactive user, we sort the similarity values in descending order and choose the top-10 most similar active users to obtain the similar users set $G = \{u_{a1}, u_{a2}, \cdots, u_{a10}\}$.

(3) Get the recent check-in sequences of top-10 similar users, and combine them into a large track sequence $C_G^{tw} = C_{u_{a1}}^{tw} \cup C_{u_{a2}}^{tw} \cup C_{u_{a3}}^{tw} \cup \cdots \cup C_{u_{a10}}^{tw}$. Then, the sequence C_G^{tw} is filled into the recent track sequence $C_{u_{in}}^{tw}$ of the inactive user u_{in}, and a new recent check-in sequence $C_{u_{in}}^{tw} = C_{u_{in}}^{tw} \cup C_G^{tw}$ is generated. Specifically, in recent sequences integration phase, the recent sequence of user u_{in} is placed in front of the sequence, then sort sequences in descending order according to repetitions of the same sequence, and delete duplicate sequences.

For each user in G, we obtaine the recent check-in sequences and combine them into a large tracking sequence, $C_G^{tw} = C_{u_{a1}}^{tw} \cup C_{u_{a2}}^{tw} \cup C_{u_{a3}}^{tw} \cup \cdots \cup C_{u_{a10}}^{tw}$. Further, we fill the sequence C_G^{tw} into the recent track sequence $C_{u_{in}}^{tw}$ of the inactive user u_{in} and generate a new recent check-in sequence $C_{u_{in}}^{tw} = C_{u_{in}}^{tw} \cup C_G^{tw}$. Specifically, in the integration phase, the recent sequence of u_{in} is placed in front of the sequence $C_{u_{in}}^{tw}$; all sequences are sorted in descending order based on repetitions; and duplicate sequences are deleted.

3.4 Adaptive Short- and Long-Term Preferences Learning

On mobile social networks, the POIs based on users' continuous check-ins usually exhibit a certain correlation. Along with user interest, the continuous check-in trajectory sequence also elucidates the periodic behavior and changes in mobile trajectory. To obtain the user preferences effectively, we design an adaptive learning strategy based on adaptive sequence processing.

3.4.1 Short-Term Preferences

The user's next POI to be visited often has an important contextual relationship with recently visited locations, illustrating the importance of short-term preferences in successive POI recommendations. Short-term preferences represent users' recent interest features that change easily over time. For instance, during vacations or business trips, user interests focus on tourism-related POIs. Moreover, a closer distance between two adjacent checked-in locations indicates a higher correlation and continuity. This can be attributed to the fact that users are more likely to select nearby locations to perform activities within certain areas.

Therefore, the short-term user interests are greatly affected by time and geographical factors, which have been fully considered in the proposed model. The RNN with spatiotemporal context (ST-RNN), with the input, hidden, and output layers, is used to model recent sequences (Fig. 3). Unlike traditional RNNs, time and distance context information hidden in the sequence are integrated into the ST-RNN.

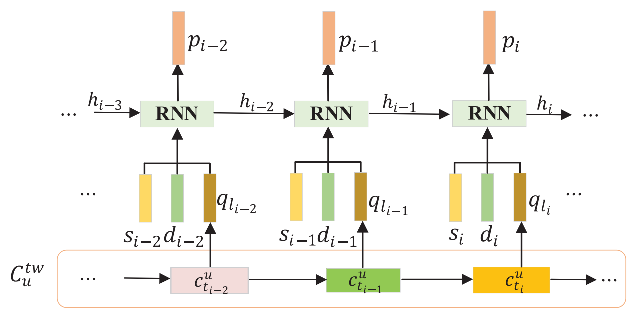


Figure 3: ST-RNN architecture for short-term preference modeling

The recent trajectory of u can be denoted as $C_u^{tw} = \left\{c_{tw_u}^u, \cdots, c_{t_i}^u, \cdots, c_{t_c}^u\right\}$. POIs are extracted from each check-in record in turn to obtain the sequence of locations $l_{tw_u} \rightarrow \cdots \rightarrow l_{t_i} \rightarrow \cdots \rightarrow l_{t_c}$. For any two adjacent check-in records $c_{t_{i-1}}^u$ and $c_{t_i}^u$, the time interval can be calculated as $\Delta t_i = t_i - t_{i-1}$, and the distance between the two adjacent visited locations l_i and l_{i-1} can be calculated by using their longitudes and latitudes as

$$\Delta d_i = R * \arccos[sinLat_i \times sinLat_{i-1} + cosLat_i \times cosLat_{i-1} \times \cos(Lon_{i-1} - Lon_i)] \tag{4}$$

where Lon_i and Lat_i denote the longitude and latitude of the location l_i, respectively; and $R = 6371$ km is the radius of the earth.

Three types of information enter the input layer: (i) current POI information, (ii) check-in time interval from the previous location, and (iii) distance from the previous location. The state of every node at each instance is not only related to the output state at the previous instance and input state of the current instance, but is also associated with the distance from and time interval to the previous POI. Thus, the proposed model reflects the impact of spatiotemporal contexts on user preferences. The status update is performed as follows:

$$h_i = \sigma(Uq_{l_i} + W_s s_i + W_d d_i + Wh_{i-1}) \tag{5}$$

$$p_i = g(V_s h_i) \tag{6}$$

where $h_i \in \mathbb{R}^d$ is the hidden layer state, which is a memory unit of the network and can be transferred to the next instance; h_i is a d-dimension feature vector, that denotes updated model information after inputting a POI l_i, which is used to record the feature information of recent trajectory; $q_{l_i} \in \mathbb{R}^d$ is the embedding representation of l_i (a model input); $s_i \in \mathbb{R}^d$ is the feature vector of the time interval Δt_i; $d_i \in \mathbb{R}^d$ is the feature vector of the distance Δd_i; $p_i \in \mathbb{R}^d$ is the model output, i.e., the user preference after inputting POI l_i; $U, W_d, W_s, W \in \mathbb{R}^{d\times d}$ are the relative weight matrices; $V_s \in \mathbb{R}^{d\times d}$ is the weight matrix of the output; and σ and g are the sigmoid and tanh activation functions, respectively.

With complete user check-in trajectories as the input, the ST-RNN model can be used to learn the recent trajectories and finally output the user's short-term interest feature, p_i, for successive POI recommendations.

3.4.2 Long-Term Preferences

Long-term interests indicate users' consistent and stable preference characteristics that do not change easily. Most users usually have stable lifestyles, which are often manifested in the form of periodic/long-term location-visiting modes. For example, users who like fitness will visit the gym and sports center regularly every week. Therefore, long-term preferences are also important for successive POI recommendations.

With the network operating over extended periods, the users' historical trajectory sequences become increasingly longer which traditional RNNs cannot handle appropriately. To effectively obtain long-term characteristics, we instead adopt an LSTM network to model historical trajectories. The forgetting gate of LSTM filters out certain unimportant feature information in the historical check-in track sequences, while hidden cell units retain long-term stable characteristics. Unlike ordinary sequential data, user interests and check-in sequences are dynamic. The earlier the check-in records, the more difficult it is to accurately reflect user interests. Therefore, the impact of time on long-term preferences should be considered when modeling historical track sequences. However, long-term interests are less affected by geography, and hence, we do not consider the influence of spatial context in this study. Therefore, we established an LSTM model by integrating the temporal factor (T-LSTM; Fig. 4).

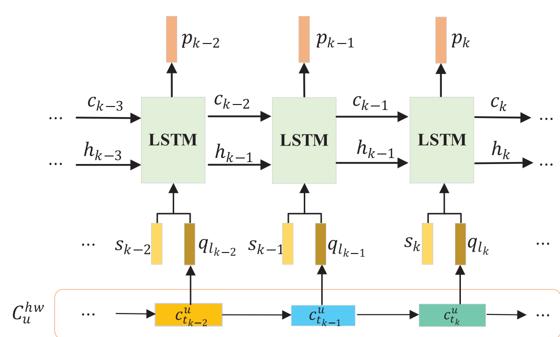


Figure 4: T-LSTM model architecture

Before modeling long-term interests, we extract the sequence of check-in locations $l_1 \to \cdots \to l_{t_k} \to \cdots \to l_{twu}$ from $C_u^{hw} = \{c_{t_1}^u, \cdots, c_{t_k}^u, \cdots, c_{twu}^u\}$, and then, calculate the time intervals of adjacent check-in records, $\Delta t_k = t_k - t_{k-1}$. Two types of information enter the input layer: POI information and check-in time interval. We denote the embedding representation vector of POI l_k as $q_{l_k} \in \mathbb{R}^d$ and the feature vector of time interval Δt_k as $s_k \in \mathbb{R}^d$.

The new candidate state after each input is only relevant to the current input location and the state passed from the previous step. Therefore, candidate state $\widetilde{c_k}$ can be updated as

$$\widetilde{c_k} = \tanh(W_c[p_{k-1}, q_{l_k}] + b_c) \tag{7}$$

Candidate state $\widetilde{c_k}$ is only used to indicate the user's current interests, while the memory cell $c_k \in \mathbb{R}^d$ retain the historical check-in location information to reflect long-term preferences. In traditional LSTM models, forget and input gates control the updating of c_k. Specifically, the input gate controls which of the users' current interests can be regarded as long-term interests, while the forget gate only distinguishes between the information that is to be retained in or discarded from the long-term interests, without considering their decay characteristics over time. Therefore, a time gate is added to the T-LSTM model to control the accumulation speed of long-term preferences. In this manner, some early interests of the users are gradually attenuated, while long-term stable preferences are retained. State c_k can then be updated according to the previous state and current location as

$$c_k = T_k \odot f_k \odot c_{k-1} + i_k \odot \widetilde{c}_k \tag{8}$$

where $i_k, f_k \in \mathbb{R}^d$ are the input and forget gates, respectively, defined as

$$i_k = \sigma(W_i[p_{k-1}, q_{l_k}] + b_i) \tag{9}$$

$$f_k = \sigma(W_f[p_{k-1}, q_{l_k}] + b_f) \tag{10}$$

and $T_k \in \mathbb{R}^d$ is a time gate that can control the attenuation of long-term interests according to the input location and time context; it can be expressed as

$$T_k = \sigma(W_q q_{l_k} + \sigma(W_t s_k) + b_t) \tag{11}$$

where $W_c, W_i, W_f \in \mathbb{R}^{d\times 2d}$ and $W_q, W_t \in \mathbb{R}^{d\times d}$ are the weight matrixes; $b_i, b_f, b_c, b_t \in \mathbb{R}^d$ are the bias vectors of the LSTM units; and $\odot$ denotes the dot product. The forget gate filters long-term interests and the time gate also filters certain interests that have not been updated in the early stage, maintaining the user's long-term and stable interests.

Once C_u^{hw} is learned, the cell state c_k of the model is obtained, representing the users' long-term preferences. Unlike traditional LSTM outputs, the proposed method only requires long-term accumulated preferences of the user, $P_k \in \mathbb{R}^d$, expressed as

$$p_k = \tanh(V_l c_k) \tag{12}$$

where $V_l \in \mathbb{R}^{d\times d}$ is a learnable model parameter.

3.5 Adaptive Successive POI Recommendation

After obtaining the long short-term interests, the predicted values of locations are calculated for recommendation. Given the vast number of locations in LBSNs, it would be computationally expensive to calculate the probability values of all locations. Therefore, we select a radius of 20 km from the user's current location to find candidate locations for recommendations based on the proportion of users checking in at POIs within that range in the Gowalla (80%) and Foursquare (99%) datasets according to a previous distance analysis [30]. The set of candidate POIs can be expressed as

$$L_{l_c} = \{l_c \in L, d_{l_c,l_o} \le 20\} \tag{13}$$

The predicted probability, S_{u,l_c}, of the candidate POI l_c is calculated using the inner product of short and long representations as

$$S_{u,l_c} = (p_i + p_k)^T q_{l_c} \tag{14}$$

where $q_{l_c} \in \mathbb{R}^d$ is a d-dimension vector, which is the feature-embedded representation of candidate POI l_c. Subsequently, the top-n POIs are recommended to the users.

4 Experiments and Analysis

4.1 Datasets and Setup

In this paper, we use two large-scale LBSNs check-in datasets as experimental data: Foursquare and Gowalla. The detailed statistical data of two datasets are listed in Table 2. We can see that the two datasets have different scales, which is more effective verification for testing the performance of methods. Furthermore, the check-in datasets are very sparse, resulting in low performance of recall and precision. For each user, the records in a dataset are arranged in ascending order of check-in time. The former 84% of check-in records are used as training data, and the remaining records are used as testing data.

Table 2: Statistics of two LBSNs check-in datasets

Items	Foursquare	Gowalla
.txt file size (MB)	11.8	25.7
# Check-ins	194,108	456,905
# Locations	5596	24,236
# Users	2321	10,162
# Avg.check-ins of each POI	34.69	18.85
# Avg.visited users of each POI	18.90	12.72
# Avg.check-ins of each user	83.63	44.97
# Avg.checked-in POIs of each user	45.57	30.26
User-POI matrix density	14.90×10^{-3}	1.86×10^{-3}
Data sparsity	98.51%	99.81%
Periods of collected data (month)	12	21

In this study, we use the Foursquare and Gowalla large-scale LBSN check-in datasets (Table 2). The two datasets are of different scales, which allows for an effective verification of model performance. Furthermore, the check-in datasets are very sparse, resulting in low recall and precision values. For each user, the records of both datasets are arranged in the ascending order of check-in time and split into a training-testing ratio of 84:16.

The NN model experiments for the proposed method are run on a GPU server with an Intel i9-9900X processor (3.5 GHz, 10 cores, and 20 threads) and two MSI GeForce RTX 2080 Ti 11G GPUs (64 GB). The operating system of the server is Ubuntu 19.04 (64-bit). The programming environment of the experimental code is Python 3.7.3, with TensorFlow 1.10.1 used as the machine learning framework.

4.2 Evaluation Metrics

We use precision, recall, and F-measure to evaluate the performance of the POI recommendation methods. Formally, given a target user $u \in U$, $R_u@n$ represents the set of top-n recommended POIs and $T_u@n$ represents the set of corresponding ground truth POIs in the testing data.

Precision measures the accuracy of the POI recommendation algorithms; it can be defined as the ratio of the number of locations actually visited to the total number of recommended POIs, and is expressed as

$$precision\ @\ n = \frac{\sum_{u \in U} |T_u @\ n \cap R_u @\ n|}{\sum_{u \in U} |R_u @\ n|} \tag{15}$$

Recall measures the comprehensiveness of an algorithm; it is defined as the ratio of the number of locations actually visited to the total number of locations in the testing set, and can be expressed as

$$recall\ @\ n = \frac{\sum_{u \in U} |T_u @\ n \cap R_u @\ n|}{\sum_{u \in U} |T_u @\ n|} \tag{16}$$

F-measure is the weighted harmonic mean of recall and precision, which is used to comprehensively evaluate the performance of a POI algorithm. It can be expressed as

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \times recall}{\beta^2 \cdot precision + recall} \tag{17}$$

where $\beta = 1$ means that recall is as important as precision.

4.3 Comparative Methods

We compare the performance of LSTP-ASR with that of five other successive POI recommendation algorithms on the Gowalla and Foursquare datasets (Table 3).

Table 3: POI recommendation methods for comparison

Algorithms	Factors	Description
FPMC-LR	Sequence and geographical influences	Local region constraint and Markov chain-based personalized factorization method [11]
STELLAR	Temporal and geographical influences	POI recommendation based on spatiotemporal potential ranking tensor decomposition [14]
NEXT	Temporal and geographical influences	A neural network frame with DeepWalk and multiple factors [20]
ST-LSTM	Spatial, temporal and sequence influences	A neural network model with spatial-temporal factor and LSTM [25]
RTPM	Temporal influence, POI category	A uniform framework based on LSTM and temporal factor [28]

(Continued)

Table 3 (continued)

Algorithms	Factors	Description
LSTP-ASR	Spatial, temporal and sequence influences, user activity, long-short term preferences	Adaptive processing of check-in sequence, adaptive learning of short and long-term interests via RNN, LSTM and spatiotemporal context (Section 3)

4.4 Results and Discussions

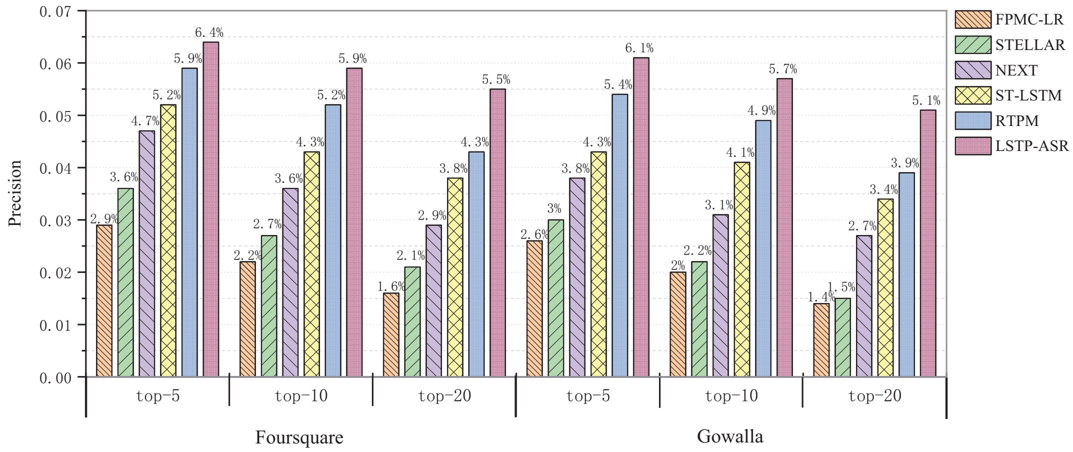
4.4.1 Performance Comparison

We record the precision, recall, and F_β-measure of the six recommendation methods for the top-n values on the two datasets (Fig. 5; Table 4). Based on our results, we made the following observations:

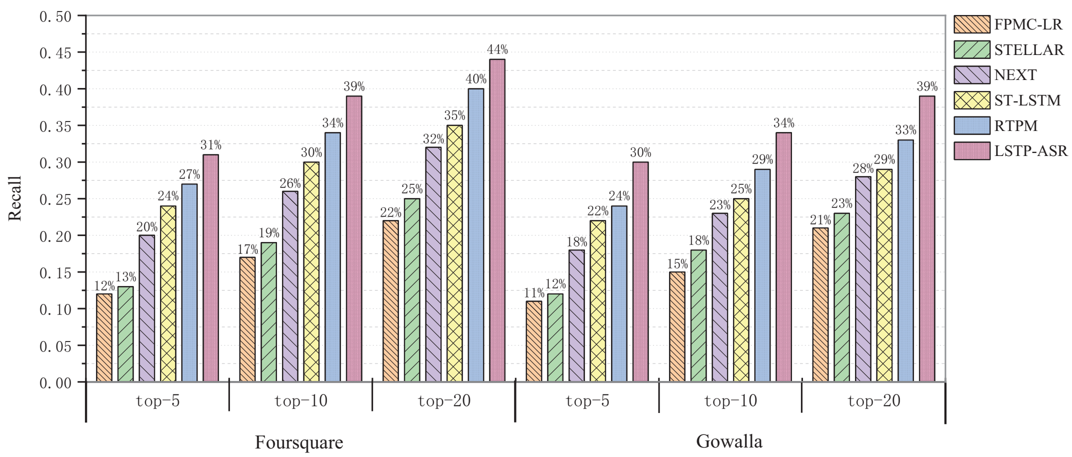
(1) The proposed LSTP-ASR model outperforms the other algorithms in terms of all three metrics on both datasets. For LSTP-ASR, on the Foursquare dataset: (i) the top-5 precision increases by 118.77%, 76.58%, 36.67%, 23.51%, and 8.83%; and (ii) the top-20 recall increases by 96.85%, 75.50%, 36.14%, 26.67%, and 10.35%, compared to that of FPMC-LR, STELLAR, NEXT, ST-LSTM, and RTPM, respectively. On the Gowalla dataset: (i) the top-5 precision increases by 135.14%, 101.66%, 59.84%, 40.97%, and 13.62%; and (ii) the top-20 recall increases by 85.65%, 67.97%, 40.58%, 34.26%, and 17.22% compared to the five baseline methods, respectively (Fig. 5). Similarly, LSTP-ASR outperforms FPMC-LR, STELLAR, NEXT, and ST-LSTM by almost (i) 126.67%–230.84%, 88.79%–155.71%, 39.91%–86.33%, and 24.77%–45.13%, respectively, on the Foursquare dataset for the top-n F_β-scores; and (ii) 142.52%, 108.53%, 60.69%, and 40.04%, respectively, for top-5 F_β-scores on the Gowalla dataset (Table 4). Importantly, LSTP-ASR outperforms the best baseline method, i.e., RTPM, with a 17.15% and 20.62% average improvement on the two datasets, respectively, in terms of F_β-scores. These improved results for LSTP-ASR can be attributed to the integration of both long-term stable and short-term preferences of the users. The short-term interests reflect the changes in the user interests, which can accurately reflect the selection of the next POI. Simultaneously, LSTP-ASR also provides adaptive check-in sequence processing, which can adapt to different types of users and achieve better recommendation performance.

(2) On both datasets for each algorithm, the recall increases and precision decreases as we increase the number of top-n predictions. For example, for top-5, top-10, and top-20 predictions, LSTP-ASR on the Foursquare dataset reaches a (i) recall of 31.3%, 38.6%, and 43.7%, respectively, and (ii) precision of 6.41%, 5.94%, and 5.53%, respectively. The reasoning for these results is as follows: (i) In the expression for recall, the denominator is a constant. Thus, increasing the number of accurately recommended POIs can increase the recall for larger top-n; and (ii) based on the precision definition, with the increasing top-n value, the increase in the numerator is not as significant as that in the denominator, which leads to a decrease in the precision.

(3) Under similar conditions, the results of all methods on the Foursquare dataset are higher than those on the Gowalla dataset. The larger and sparser nature of the latter make user check-in behavior and POIs more complex and diverse, leading to less accurate recommendations for every algorithm.



(a)



(b)

Figure 5: Performance comparison of different successive POI recommendation algorithms based on (a) precision and (b) recall metrics for top-*n* predictions on the Foursquare and Gowalla datasets

Table 4: F_β-measure ($\beta = 1$) on two datasets. The best performance scores are in bold

Datasets	Metrics	FPMC-LR	STELLAR	NEXT	ST-LSTM	RTPM	LSTP-ASR
Foursquare	top-5	0.0469	0.0564	0.0761	0.0853	0.0968	**0.1064**
	Improved	126.67%	88.79%	39.91%	24.77%	9.89%	
	top-10	0.0386	0.0472	0.0624	0.0759	0.0899	**0.1030**
	Improved	166.42%	118.04%	65.12%	35.66%	14.48%	
	top-20	0.0297	0.0384	0.0527	0.0676	0.0773	**0.0982**
	Improved	230.84%	155.71%	86.33%	45.13%	27.09%	
Gowalla	top-5	0.0416	0.0484	0.0628	0.0721	0.0877	**0.1010**
	Improved	142.52%	108.53%	60.69%	40.01%	15.12%	
	top-10	0.0356	0.0397	0.0553	0.0707	0.0839	**0.0980**

(Continued)

Table 4 (continued)

Datasets	Metrics	FPMC-LR	STELLAR	NEXT	ST-LSTM	RTPM	LSTP-ASR
	Improved	175.11%	146.67%	77.41%	38.69%	16.79%	
	top-20	0.0254	0.0287	0.0494	0.0600	0.0696	**0.0905**
	Improved	256.69%	215.21%	83.29%	50.67%	29.94%	

4.4.2 Analysis of Key Components in LSTP-ASR

To further investigate the effectiveness of key components in the proposed model, we conduct ablation experiments on the following three variants of LSTP-ASR (Fig. 6):

- LSTP-ASR-V1: This variant uses only an LSTM model to learn user preference instead of the adaptive learning of user short-term interest preferences (Section 3.4). The rest of the framework is preserved.
- LSTP-ASR-V2: To verify its efficacy, adaptive processing of the check-in sequence is removed (Section 3.3), while the rest of the framework is preserved. In this variant, only one recently checked-in location is used for short-term interests.
- LSTP-ASR-V3: To verify the impact of distance and time context on model performance, we delete the (i) distance feature vector, d_i, and time interval vector, s_i, from the ST-RNN (Eq. (5)), and (ii) the time interval feature vector, s_k, from the T-LSTM (Eq. (11)). The rest of the framework is preserved.

The following conclusions are drawn from the experimental results:

- The complete LSTP-ASR model achieves the best performance for both datasets for given parameters compared to all other versions, indicating the key components to be important for effective successive POI recommendation.
- LSTP-ASR-V2 outperforms LSTP-ASR-V1 demonstrating that the adaptive learning of short-term preferences better mined user features, substantially improving successive POI recommendations. As we know, user preference learning is indeed the most crucial part of the POI recommendation process.
- Adaptive sequence processing is found to be an indispensable factor; without it, the precision and recall of LSTP-ASR decrease by 0.5% and 2.6% on average for the Foursquare and Gowalla datasets, respectively. Hence, the inclusion of adaptive sequence processing is necessary as it better generates long- and short-term check-in sequences.
- The performance of LSTP-ASR-V3 is inferior to that of LSTP-ASR, suggesting that spatiotemporal context enhances POI recommendation performance. Long short-term preferences are closely related to the time features, while the distance features reflect short-term movement patterns.

Therefore, all three key components of LSTP-ASR effectively improve the POI recommendation performance.

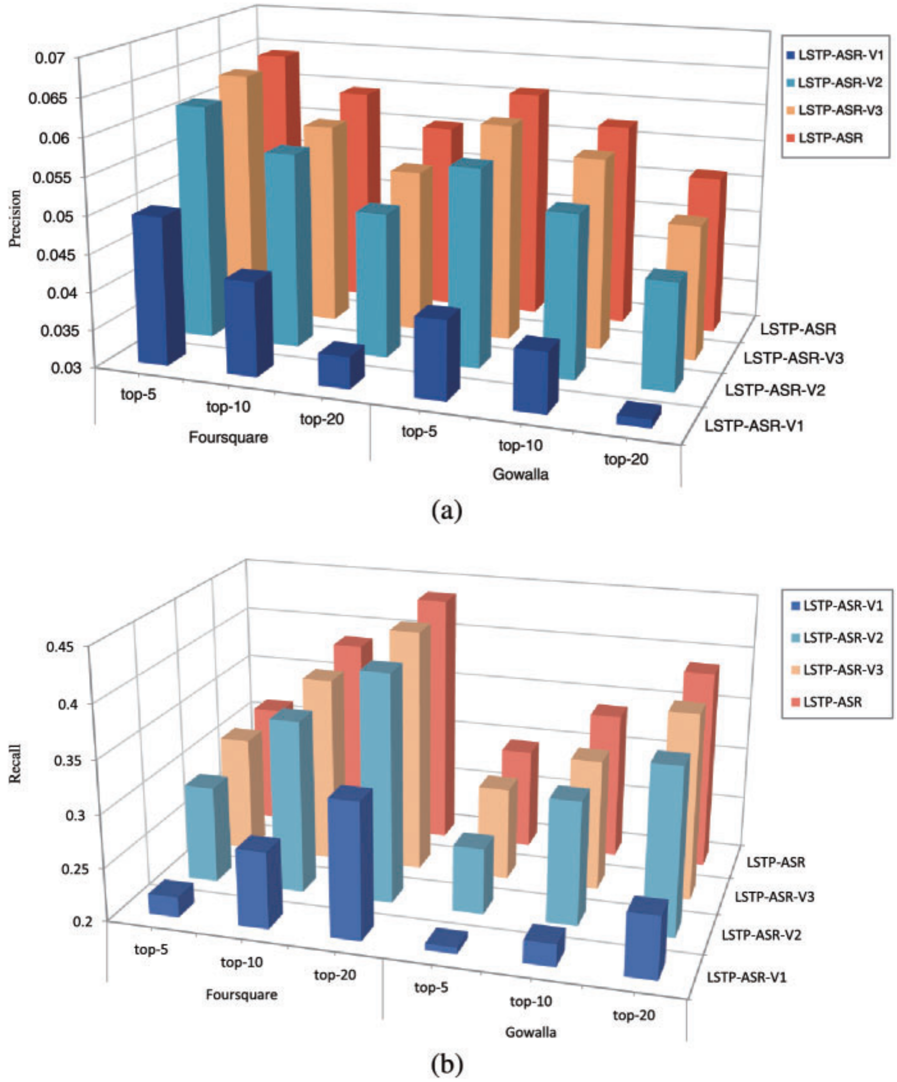


Figure 6: (a) Precision and (b) recall @ top-*n* for the different LSTP-ASR variants on the Foursquare and Gowalla datasets

4.4.3 Impact of Minimum Check-in Sequence Length

The adaptive sequence filling strategy of LSTP-ASR requires a check-in sequence length parameter, δ_{min}, to divide inactive and active users. We analyze the effect of a range of values for $\delta_{min} \in \{10, 20, 30, 40, 50\}$ to evaluate its influence on model performance by computing the top-5 precision and recall values on both datasets (Fig. 7). LSTP-ASR achieves the best performance for $\delta_{min} = 20$ and 40 on the Gowalla and Foursquare datasets, respectively. This may be attributed to weaker user correlations for sparser datasets, which degrades the performance upon simply increasing the sequence length. Therefore, the adaptive sequence length benefits the proposed LSTP-ASR model.

4.4.4 Generalizability of LSTP-ASR

To validate the generalizability of LSTP-ASR, we conduct experiments using the Brightkite dataset for different regions with varying user behavior patterns. Foursquare is the check-in dataset for users in Singapore and Gowalla is that for users in California and Nevada, USA. However, Brightkite is a larger check-in dataset for users around the world; it includes 50,687 users, 702,401 POIs, and 4,452,694 check-ins, and has a sparsity of 99.98%.

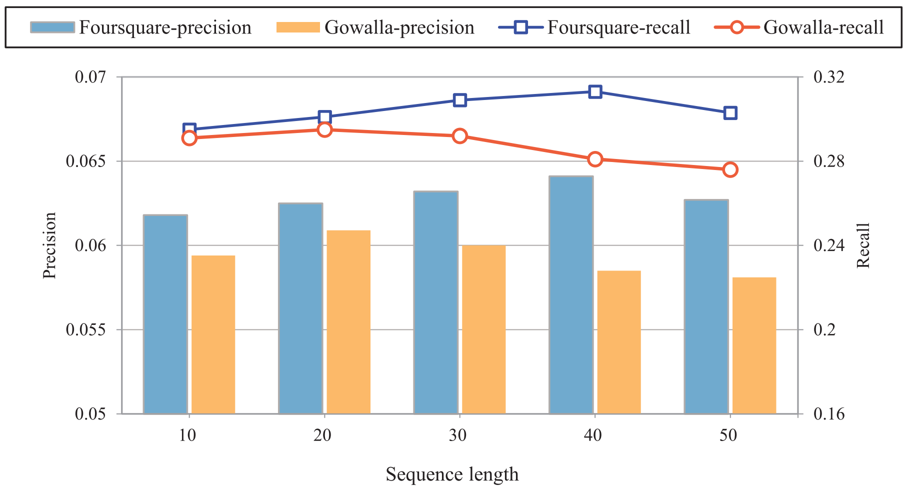


Figure 7: Impact of the sequence length parameter, $\boldsymbol{\delta}_{\mathbf{min}}$, on LSTP-ASR

Furthermore, we divide the Brightkite dataset at two scales to study the model performance under different dataset sizes: (i) Brightkite-50%, i.e., the first half of the Brightkite dataset with earlier check-in times; and (ii) Brightkite-100%, i.e., the entire Brightkite dataset. We use LSTP-ASR for performance evaluation (Table 5) and observe the following: (i) The precision, recall, and F_1-measure on Brightkite are slightly lower than those on Foursquare and Gowalla due to the former being the sparsest dataset, for which accurate POI recommendation is more difficult; and (ii) as the dataset size increased, all three metric results show gradual upward trends. The larger amount of check-in records helps the model training, but performance improvement is still limited due to sparsity. These results validate the generalizability of the proposed LSTP-ASR model for different LBSN check-in datasets of varying sizes.

Table 5: Performance of different dataset sizes on Brightkite

Metrics	Brightkite-50%			Brightkite-100%		
	top-5	top-10	top-20	top-5	top-10	top-20
Precision	0.0587	0.0541	0.0484	0.0601	0.0562	0.0504
Recall	0.282	0.315	0.343	0.291	0.324	0.368
F_1-measure	0.0972	0.0923	0.0848	0.0996	0.0958	0.0887

5 Discussion

LSTP-ASR is an adaptive successive POI recommendation model, which can effectively solve the problems of inaccurate user preference extraction and cold start for inactive users. The core modules of LSTP-ASR are adaptive check-in sequence processing and adaptive long short-term preferences learning for different types of users. Therefore, the LSTP-ASR model is suitable for recommending locations in various practical scenarios:

1. Diverse user needs: LSTP-ASR can provide adaptive and personalized POI recommendations, which is suitable for both active users with more check-in records and inactive users with shorter check-in trajectory sequences.
2. Inactive user cold start: For new users with only a few check-in records, LSTP-ASR can find similar active users with common preferences and recommend locations visited by them to the new users (e.g., recommending POIs to office workers who have regular lifestyle and work schedule).

The implementation of the proposed method is similar to the existing recommendation systems. Once the algorithm is programmed and run using Python, it can be deployed in the existing LBSN servers without additional equipment or configuration. On the LBSN platforms, the user check-in records are uploaded to the LBSN servers in real-time through the check-in function. LSTP-ASR regularly runs model training to learn user preferences and provides LBSN users with the top-n recommended POIs in real time.

The practical significance of LSTP-ASR is reflected in three aspects: (i) For users, the proposed method can effectively reduce the selection confusion caused by location information overload and assist users in exploring new locations to enhance their experiences, especially when checking out previously unvisited areas; (ii) For businesses, merchants can explore potential users and send coupons/advertisements to those who have checked-in at the stores to improve business benefits; (iii) For recommendation systems, the proposed method can promote the application and development of social networks, smart cities, and intelligent recommendations. In conclusion, the proposed LSTP-ASR successive POI recommendation method has important and useful practical value in LBSN application services.

6 Conclusions and Future Work

The current successive POI recommendation systems lack adaptive learning of user preferences, often leading to inaccurate reflections. In this study, we propose the LSTP-ASR adaptive successive POI recommendation method with trajectory sequence processing and long short-term preference learning. This model adopts two core strategies: adaptive check-in sequence processing and adaptive long short-term preference learning. An adaptive sequence filling strategy is used to expand the recent check-in records of inactive users using those of similar active users to accurately infer their short-term preferences. For recent and historical check-in trajectory sequences, RNN and LSTM models with spatiotemporal and temporal context, respectively, are used to adaptively learn the short and long preferences of users. The proposed model can solve the problems of inaccurate extraction of user preferences and cold start for inactive users. The experimental results on the Foursquare and Gowalla datasets indicate that LSTP-ASR outperforms other baseline POI recommendation models in terms of precision, recall, and F_β-measure. In the future, we will investigate the influence of weather, fine-grained user features, and time characteristics (e.g., workdays, weekends, and holidays) to further improve the recommendation performance.

Acknowledgement: Professor Shengrong Gong and Professor Xiaoshuang Xing from Changshu Institute of Technology provide valuable guidance on the conceptualization and methodology of our work. The experimental resources and computing environment are provided by School of Computer Science and Engineering, Changshu Institute of Technology. We are thankful for their support.

Funding Statement: This research was funded by the National Natural Science Foundation of China (Grant Nos. 62102347, 62376041, 62172352) and Guangdong Ocean University Research Fund Project (Grant No. 060302102304).

Author Contributions: Yali Si: Conceptualization, Methodology, Research design, Writing—original draft, Writing—review & editing, Supervision. **Feng Li:** Investigation, Conceptualization, Validation, Writing—original draft, Writing—review & editing. **Shan Zhong:** Supervision, Methodology, Writing—original draft. **Chenghang Huo:** Investigation, Software, Validation, Writing—original draft. **Jing Chen:** Validation, Methodology, Writing—review & editing. **Jinglian Liu:** Formal analysis, Data curation, Validation. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: We used publicly available data and gave a reference to it in our paper. The data used to support the findings of this study are available from the corresponding author upon request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. Chen, W. Jiang, J. Wu, and K. Li, "Dynamic personalized POI sequence recommendation with fine-grained contexts," *ACM Trans. Internet Technol.*, vol. 23, no. 2, pp. 1–28, 2023. doi: 10.1145/3583687.

[2] M. Davtalab and A. A. Alesheikh, "A POI recommendation approach integrating social spatio-temporal information into probabilistic matrix factorization," *Knowl. Inf. Syst.*, vol. 63, no. 1, pp. 65–85, 2021. doi: 10.1007/s10115-020-01509-5.

[3] Y. L. Si, F. Z. Zhang, and W. Y. Liu, "CTF-ARA: An adaptive method for POI recommendation based on check-in and temporal features," *Knowl. Based Syst.*, vol. 128, no. 1, pp. 59–70, 2017. doi: 10.1016/j.knosys.2017.04.013.

[4] K. Seyedhoseinzadeh, H. Rahmani, M. Afsharchi, and M. Aliannejadi, "Leveraging social influence based on users activity centers for point-of-interest recommendation," *Inf. Process. Manag.*, vol. 59, no. 2, 2022, Art. no. 102858. doi: 10.48550/arXiv.2201.03450.

[5] Y. Li, F. Yin, and X. Hui, "Recommendation algorithm integrating CNN and attention system in data extraction," *Comput. Mater. Contin.*, vol. 75, no. 2, pp. 4047–4062, 2023. doi: 10.32604/cmc.2023.036945.

[6] Q. Yi, H. Gao, X. Chen, and X. Kong, "Human mobility pattern prior knowledge based POI recommendation," *Comput. Sci.*, vol. 50, no. 9, pp. 139–144, 2023. doi: 10.11896/jsjkx.220900114.

[7] R. Li, X. Meng, and Y. Zhang, "A points of interest recommendation framework based on effective representation of heterogeneous nodes in the Internet of Things," *Comput. Commun.*, vol. 196, no. 16, pp. 76–88, 2022. doi: 10.1016/j.comcom.2022.09.014.

[8] Z. B. Zhang, C. Zou, R. F. Ding, and Z. Chen, "VCG: Exploiting visual contents and geographical influence for point-of-interest recommendation," *Neurocomputing*, vol. 357, no. 4, pp. 53–65, 2019. doi: 10.1016/j.neucom.2019.04.079.

[9] B. Chang, Y. Park, D. Park, S. Kim, and J. Kang, "Content-aware hierarchical point-of-interest embedding model for successive POI recommendation," in *Proc. IJCAI*, Stockholm, Sweden, 2018, pp. 3301–3307.

[10] Z. Huang, H. Ma, S. Wang, and Y. Shen, "Accurate item recommendation algorithm of itemrank based on tag and context information," *Comput. Commun.*, vol. 176, no. 1, pp. 282–289, 2021. doi: 10.1016/j.comcom.2021.06.020.

[11] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *Proc. IJCAI*, Beijing, China, 2013, pp. 2605–2611.

[12] S. Feng, X. Li, Y. Zeng, G. Cong, and Y. M. Chee, "Personalized ranking metric embedding for next new POI recommendation," in *Proc. IJCAI*, Buenos Aires, Argentina, 2015, pp. 2069–2075.

[13] J. He, X. Li, L. Liao, D. Song, and W. K. Cheung, "Inferring a personalized next point-of-interest recommendation model with latent behavior patterns," in *Proc. AAAI*, Phoenix, AZ, USA, 2016, pp. 137–143.

[14] S. Zhao, T. Zhao, H. Yang, M. R. Lyu, and I. King, "STELLAR: Spatial-temporal latent ranking for successive point-of-interest recommendation," in *Proc. AAAI*, Phoenix, AZ, USA, 2016, pp. 315–321.

[15] P. Chen, W. Shi, X. Zhou, Z. Liu, and X. Fu, "STLP-GSM: A method to predict future locations of individuals based on geotagged social media data," *Int. J. Geogr. Inf. Sci.*, vol. 33, no. 12, pp. 2337–2362, 2019. doi: 10.1080/13658816.2019.1630630.

[16] H. Ying, J. Wu, G. Xu, Y. Liu, T. Liang and X. Zhang, "Time-aware metric embedding with asymmetric projection for successive POI recommendation," *World Wide Web*, vol. 22, no. 5, pp. 2209–2224, 2019. doi: 10.1007/s11280-018-0596-8.

[17] M. Gan and Y. Ma, "Mapping user interest into hyper-spherical space: A novel POI recommendation method," *Inf. Process. Manag.*, vol. 60, no. 2, 2023, Art. no. 103169. doi: 10.1016/j.ipm.2022.103169.

[18] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proc. AAAI*, Phoenix, AZ, USA, 2016, pp. 194–200.

[19] 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 Trans. Inf. Syst.*, vol. 35, no. 4, pp. 1–28, 2017. doi: 10.1145/3041658.

[20] Z. Zhang, C. Li, Z. Wu, A. Sun, D. Ye and X. Luo, "NEXT: A neural network framework for next POI recommendation," *Front. Comput. Sci.*, vol. 14, no. 2, pp. 314–333, 2020. doi: 10.1007/s11704-018-8011-2.

[21] Y. S. Lu, W. Y. Shih, H. Y. Gau, C. K. Chieh, and J. L. Huang, "On successive point-of-interest recommendation," *World Wide Web*, vol. 22, no. 3, pp. 1151–1173, 2019. doi: 10.1007/s11280-018-0599-5.

[22] G. Cao, S. Cui, and I. Joe, "Improving the spatial-temporal aware attention network with dynamic trajectory graph learning for next point-of-interest recommendation," *Inf. Process. Manag.*, vol. 60, no. 3, 2023, Art. no. 103335. doi: 10.1016/j.ipm.2023.103335.

[23] Y. C. Chen, T. Thaipisutikul, and T. K. Shih, "A learning-based POI recommendation with spatiotemporal context awareness," *IEEE Trans. Cybern.*, vol. 52, no. 4, pp. 2453–2466, 2022. doi: 10.1109/TCYB.2020.3000733.

[24] L. Huang, Y. Ma, S. Wang, and Y. Liu, "An attention-based spatiotemporal LSTM network for next POI recommendation," *IEEE Trans. Serv. Comput.*, vol. 14, no. 6, pp. 1585–1597, 2019. doi: 10.1109/TSC.2019.2918310.

[25] P. Zhao *et al.*, "Where to go next: A spatio-temporal gated network for next POI recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2512–2524, 2020. doi: 10.1109/TKDE.2020.3007194.

[26] L. Li, Y. Liu, J. Wu, L. He, and G. Ren, "Multi-modal representation learning for successive POI recommendation," in *Asian Conf. Mach. Learn.*, Nagoya, Japan, 2019, pp. 441–456.

[27] H. Wang, P. Li, Y. Liu, and J. Shao, "Towards real-time demand-aware sequential POI recommendation," *Inf. Sci.*, vol. 547, no. 8, pp. 482–497, 2021. doi: 10.1016/j.ins.2020.08.088.

[28] X. Liu, Y. Yang, Y. Xu, F. Yang, Q. Huang and H. Wang, "Real-time POI recommendation via modeling long- and short-term user preferences," *Neurocomputing*, vol. 467, no. 6, pp. 454–464, 2022. doi: 10.1016/j.neucom.2021.09.056.

[29] Z. Liu, L. Meng, Q. Sheng, D. Chu, J. Yu and X. Song, "POI recommendation for random groups based on cooperative graph neural networks," *Inf. Proc. Manag.*, vol. 61, no. 3, 2024, Art. no. 103676. doi: 10.1016/j.ipm.2024.103676.

[30] Y. L. Si, F. Z. Zhang, and W. Y. Liu, "An adaptive point-of-interest recommendation method for location-based social networks based on user activity and spatial features," *Knowl. Based Syst.*, vol. 163, no. 1, pp. 267–282, 2019. doi: 10.1016/j.knosys.2018.08.031.