Computing and Informatics, Vol. 45, 2026, 297–319, doi: 10.31577/cai_2026_2_297

POINT OF INTEREST RECOMMENDATION METHOD BASED ON SPATIOTEMPORAL FEATURE DEPENDENCE

Yuan Cheng, Yu Cheng, Ao Li, Feihu Jin

School of Computer Science and Technology
Harbin University of Science and Technology
Harbin 150080, China
e-mail: chengyuan@hrbust.edu.cn, 2220400051@stu.hrbust.edu.cn, {ao.li, fhjin}@hrbust.edu.cn

Abstract. With the continuous development of smart mobile devices, user check-in behaviors data has experienced explosive growth. To solve the problem that existing point-of-interest (POI) recommendation models fail to fully consider the complex and dynamic changes in user preference behaviors, a POI recommendation model based on a Bi-directional Long Short-Term Memory recurrent neural network (Bi-LSTM) and an attention mechanism is proposed. By integrating the time sequence and spatial location information of user check-in behaviors into Bi-LSTM, the user's spatio-temporal feature sequential dependence is established. The attention mechanism is utilized to model historical check-in locations, derive a weight sequence, and recommend the next POI to the user based on the probability distribution of predicted POIs. Experimental results on two public datasets demonstrate that the model outperforms previous advanced POI recommendation models in multiple evaluation metrics and has better recommendation performance.

Keywords: Points of interest recommendation, Bi-LSTM, attention mechanism, user preferences, sequential dependence

Mathematics Subject Classification 2010: 97R40

1 INTRODUCTION

With the rapid development of information technology, location-based social networks (LBSN) have become a milestone in the evolution of the internet. Numerous services such as Google Maps, Yelp, and Facebook have brought significant convenience to people's lives, LBSNs bridge the gap between cyberspace and the physical world. As an important component of LBSN, POI recommendations have garnered widespread attention from researchers [1, 2]. POI recommendation can uncover users' potential personal preferences and lifestyle habits based on their historical check-in behaviors data, allowing for the recommendation of POIs that meet their personalized needs. POI recommendations help users search for points of interest more quickly, saving them considerable time. While businesses can use recommendation services to accurately advertise for commercial marketing [3], thereby increasing their revenue. In practical applications of POI recommendation, users' preference behaviors will constantly change with the passage of time and changes in geographical location. For example, on weekdays, User A tends to have lunch at a restaurant near the company, while on weekends, the user prefers to have lunch at a restaurant near home. The dynamic changes in user preference behaviors become more complex with the significant increase in check-in data volume, affecting the accuracy and operational efficiency of POI recommendation algorithms.

Early work on Points of Interest (POI) recommendation utilized common methods from sequential recommendation tasks for prediction, such as Markov chains, matrix factorization, and metric embedding. With the advancement of information technology, the volume of user check-in data has significantly increased, resulting in deep learning-based approaches demonstrating superior performance. Consequently, an increasing number of POI recommendation methods leverage deep learning to learn users' preference behaviors from their check-in records [4, 5]. While these methods have improved the performance of POI recommendations to some extent, they have not effectively considered the dynamic changes in user preference behaviors and cannot accurately learn users' preferences. To address the above issues, we propose a POI recommendation model based on Bi-LSTM and attention mechanism (PMBAM). The model utilizes Bi-LSTM to deeply mine the dynamic changes in user preference behaviors. Additionally, it employs an attention mechanism to alleviate the problems of information forgetting or redundancy that arise when dealing with long sequences.

The main contributions of our work are summarized as follows:

1. We deeply explore the hidden information within user check-in datasets, utilizing time intervals, spatial distances, and positional encodings to provide richer feature representations for user check-in sequences, thereby enhancing the model's ability to learn user preferences.

2. To better capture the dynamic changes in user preference behavior, we integrate the Bi-LSTM with an attention mechanism. This integration models bidirec-

tional contextual information to learn the complex dynamic changes in user preference behavior.

3. We conduct experiments and analyses on two publicly available real-world datasets, demonstrating that the model proposed in this paper effectively enhances the accuracy of next POI recommendations.

2 RELATED WORK

In recent years, the successful application of deep learning across various research fields has led numerous experts and scholars to propose many deep learning-based POI recommendation models. Most of these models utilize contextual information for recommendations, including geographical information, temporal information, social relationship information, textual information, and image information, etc. [6]. How to leverage these contextual information sources to provide more personalized POI recommendations has been a key focus for researchers.

Zhang et al. [7] considered geographical and temporal information, proposing a POI recommendation model based on graph neural networks. This model generates node representations by utilizing both node information and the topological structure, thereby enhancing the accuracy of POI recommendations. Hu et al. [8] proposed a POI recommendation model based on network relation representation learning, which utilizes various contextual information such as social information, geographical information, and user preferences to generate a social relationship network, uncovering hidden social relationships among users. Xu et al. [9] introduced a trust relationship measurement method that employs a matrix factorization model to simultaneously model trust relationships, user preferences, and spatiotemporal information. They also considered the impacts of item attractiveness and correlations between items on the recommendation algorithm, addressing issues related to cold start and sparsity in POI recommendation models. Wang et al. [10] employed a fusion graph convolutional network with an encoder-decoder architecture to alleviate the cold start problem encountered when utilizing graph structures. Lan et al. [11] combined various spatiotemporal features by employing location expansion algorithms and gated deep networks to explore the dependencies of users' non-continuous points of interest. Yan et al. [12] focused on the collaborative information between individual user trajectories and other users' trajectories, designing the STHGCN model by integrating hypergraph structure encoding with spatiotemporal information.

As user check-in behaviors datasets continue to grow, attention mechanisms have begun to be applied in POI recommendation. Compared to other traditional deep learning models, attention mechanisms can effectively filter important data from the dataset, alleviating issues of information forgetting or redundancy when processing long sequence data, and capturing the complex preference behaviors reflected in user check-in records. Liu et al. [1] proposed a POI recommendation model that integrates attention mechanisms with gated recurrent units, which helps mitigate the negative impact of data sparsity on personalized model training. Yang et al. [13] introduced a graph-enhanced transformer model that utilizes attention mechanisms to merge factors such as user preference behaviors and spatiotemporal information for POI recommendations, improving accuracy while addressing cold start issues. Li et al. [14] considered the spatiotemporal information contained among different nodes in the user check-in sequence graph, utilizing attention mechanisms and gated graph neural networks for POI recommendation. Liu et al. [15] proposed the ABG-poic model, which utilizes bidirectional gated recurrent units and an attention mechanism to predict the next category of users' POIs. Kasgari et al. [16] extracted users' latent features by mapping user check-in trajectories into different shapes, subsequently employing an attention mechanism to recommend points of interest to users.

In summary of the existing methods, it is evident that the current POI recommendation approaches mainly utilize users' historical preference behaviors and contextual information to model their preferences, improving the performance of the POI recommendation models to some extent. However, these methods do not account for the complex dynamic changes in user preference behaviors, and user preferences can vary due to different factors. Therefore, we propose a POI recommendation model that integrates Bi-LSTM and attention mechanisms. This model captures the complex dynamic changes in user preference behaviors by utilizing various contextual information along with the Bi-LSTM module. Additionally, it employs a ProbSparse attention mechanism to alleviate issues of information forgetting or redundancy that arise when the Bi-LSTM module processes longer user check-in sequences. Furthermore, compared to traditional attention mechanisms, the ProbSparse attention mechanism can focus on the more significant dot-product pairs during the learning process, which reduces the computation of similarity, enhances model efficiency, strengthens the model's ability to capture user preference behaviors, and ultimately improves the accuracy of the POI recommendation model.

3 THE PROPOSED MODEL

3.1 Model Definition

Definition 1. Check-in Records. The user set $U = \{u_1, u_2, u_3, \ldots, u_x\}$, where x denotes the total number of users in the sequence, and u_i $(1 \leq i \leq x)$ represents the i^{th} user in the dataset. The location set $P = \{p_1, p_2, p_3, \ldots, p_y\}$, where y denotes the total number of locations in the sequence, and p_i $(1 \leq i \leq y)$ represents the i^{th} location in the dataset. The individual user's time points set $T = \{t_1, t_2, t_3, \ldots, t_z\}$, where z denotes the total number of check-ins for that user, and t_i $(1 \leq i \leq z)$ represents the time point when the user checked in at the i^{th} location. The historical check-in locations of a user are sorted in chronological order, with the maximum sequence length denoted as D. The length of the user's check-in sequence B is compared with the maximum sequence length D. If B exceeds D, the first D

check-in records are retained. Conversely, if B is shorter than D, the user's check-in sequence is padded with zeros until the lengths of B and D are equal, while ensuring that the padded zeros are masked in subsequent calculations. Each check-in record $P^i_{seq_a}$ of user u_i can be represented as $[u_i, p^i_a, t^i_a]$, where p^i_a indicates the location of the a^{th} check-in by user u_i, and t^i_a indicates the time of the a^{th} check-in by user u_i. Therefore, the check-in record sequence of user u_i is expressed as $N(u_i) = \left[P^i_{seq_1}, P^i_{seq_2}, P^i_{seq_3}, \ldots, P^i_{seq_D}\right]$, where $P^i_{seq_1}$ and $P^i_{seq_D}$ represent the first and last check-in records of user u_i in chronological order, respectively.

Definition 2. Spatiotemporal Interval. We utilize time intervals and spatial intervals to further capture the dynamic changes in user preference behaviors. The time interval and spatial interval between the i^{th} and j^{th} check-ins of the same user are denoted by $\triangle^t_{ij}$ and $\triangle^d_{ij}$, respectively. The spatial interval is calculated using the Haversine distance formula [17], as shown in Equations (1) and (2).

$$\triangle^t_{ij} = |t_j - t_i|, \tag{1}$$

$$\triangle^d_{ij} = Haversine(p_i, p_j). \tag{2}$$

The Haversine distance formula can accurately calculate the shortest distance between two points on a sphere, and its relatively simple computation makes it suitable for distance calculations in large-scale datasets, such as user check-in datasets. The time interval matrix and spatial interval matrix for each check-in record of user u_i with the check-in sequence $[P^i_{seq_1}, P^i_{seq_2}, P^i_{seq_3}, \ldots, P^i_{seq_D}]$ are denoted by $F(T)$ and $F(L)$, respectively.

Definition 3. Positional Encoding. We use positional encoding to add specific relative positional information to the check-in sequence, allowing for the identification and distinction of each different check-in location. This helps the model understand the relative relationships between check-in locations, thereby enhancing the model's ability to capture user preferences. The sine positional encoding for the user's i^{th} check-in is denoted as $PE_{(p_i,2f)}$, and the cosine positional encoding is denoted as $PE_{(p_i,2f+1)}$, as shown in Equations (3) and (4).

$$PE_{(p_i,2f)} = \sin\left(p_i / 10\,000^{2f/d_{model}}\right), \tag{3}$$

$$PE_{(p_i,2f+1)} = \cos\left(p_i / 10\,000^{2f/d_{model}}\right), \tag{4}$$

where f is the encoding index dimension, and d_{model} is the dimension of the output embedding space.

Problem Definition 1. POI Recommendation Algorithm. Based on the historical check-in behavior records of the current user, predict locations that the user may be interested in next. For example, given the check-in records of user $u_i \in U$ as $[P^i_{seq_1}, P^i_{seq_2}, P^i_{seq_3}, \ldots, P^i_{seq_D}]$, predict the location that the user is likely to visit at the next time step.

3.2 Model Structure

A POI recommendation model that integrates Bi-LSTM and attention mechanisms is proposed. First, the user check-in record sequence $[P^i_{seq_1}, P^i_{seq_2}, P^i_{seq_3}, \ldots, P^i_{seq_D}]$ undergoes an embedding operation, mapping the check-in sequence into a continuous vector space. Next, the embedded check-in location records, positional encoding information, and spatiotemporal interval information are fed into the Bi-LSTM module. Then, the attention mechanism module is utilized to obtain the weight matrix. Following that, the interest point matching module is used to acquire the probability distribution of the predicted POIs. Finally, the prediction results are obtained based on the probability distribution. The overall framework of the model is shown in Figure 1.

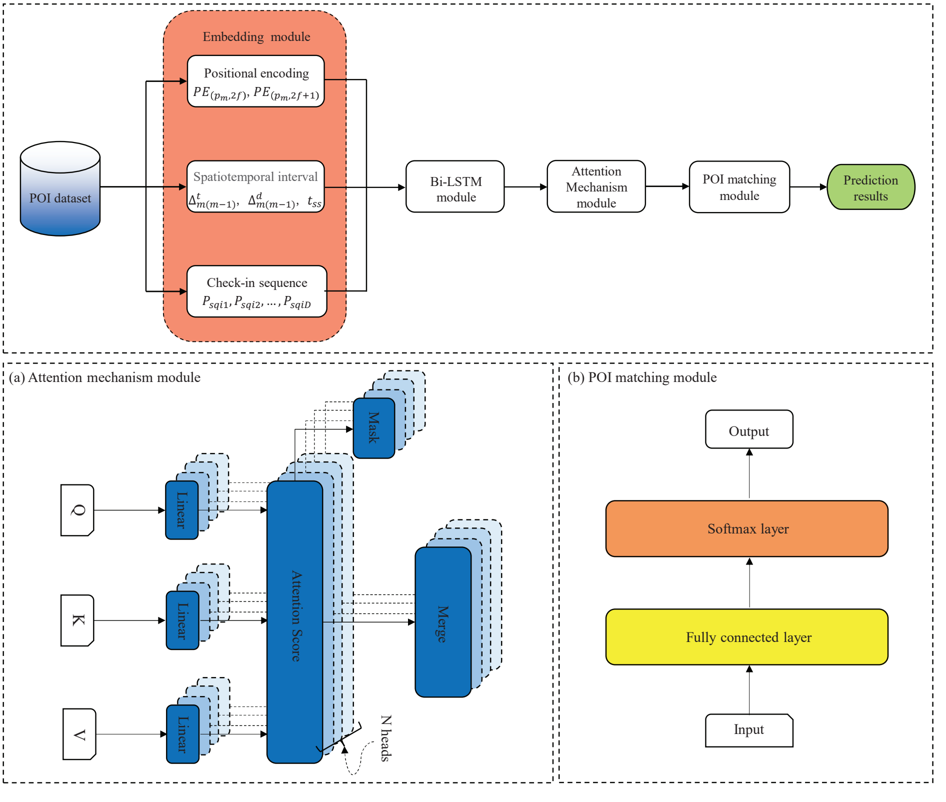

Figure 1. PMBAM model structure

3.2.1 Embedding Module

The Embedding module utilizes user check-in datasets to obtain various contextual information that includes spatiotemporal relationships. This information is integrated into subsequent modules to enhance the model's ability to capture user preference behaviors and improve the accuracy of the model's predictions. The Embedding module consists of three parts: location encoding, spatiotemporal intervals, and the check-in sequence. It combines the positional encoding, spatiotemporal interval information, and user check-in records sequence processed through embedding operations into a unified feature sequence. For user u_i's m^{th} check-in, the merged feature is represented as $a_{im} = \left(l_m, \triangle^t_{m(m-1)}, \triangle^d_{m(m-1)}, t_{ss}, PE_{(p_m,2f)}, PE_{(p_m,2f+1)}\right)$, where l_m is the location of the m^{th} check-in, $\triangle^t_{m(m-1)}$ and $\triangle^d_{m(m-1)}$ represent the time interval and spatial interval between the m^{th} check-in and the $(m-1)^{\text{th}}$ check-in, respectively. t_{ss} indicates whether it is a working day or a rest day, while $PE_{(p_m,2f)}$ and $PE_{(p_m,2f+1)}$ denote the sine and cosine positional encodings of the m^{th} check-in. The feature sequence generated from the previous D check-ins of user u_i is represented as $A_{iD} = [a_{i1}, a_{i2}, a_{i3}, \ldots, a_{iD}]$.

3.2.2 Bi-LSTM Module

User preference behaviors are not only related to historical check-in locations but also have associations with future check-in locations. Bi-LSTM is a bidirectional RNN model based on LSTM, which contains two LSTM layers that can process data in both forward and backward directions, utilizing contextual information from both directions. We use the forward LSTM to establish the relationship between user check-in locations and historical check-in locations, while the backward LSTM establishes the relationship between user check-in locations and future check-in locations. By combining bidirectional information, the model can better capture the complex dynamic changes in user preference behaviors when the dataset is large, thereby improving the accuracy of POI recommendations. The Bi-LSTM module transmits the sequences processed by the Embedding layer to the two LSTM layers and merges the outputs. The structure of the Bi-LSTM module is illustrated in Figure 2, where $x_1, x_2, \ldots, x_D$ represent the feature sequence generated by the user through check-ins from t_1 to t_D, and $Y_1, Y_2, \ldots, Y_D$ denote the corresponding final outputs at each time step, which serve as inputs for the subsequent model. The Bi-LSTM calculations are shown in Equations (5), (6), and (7).

$$h_t = \mathrm{LSTM}\left(x_t, h_{t-1}\right), \tag{5}$$

$$h'_t = \mathrm{LSTM}'\left(x_t, h'_{t-1}\right), \tag{6}$$

$$Y_t = \tanh\left(W_y h_t + W'_y h'_t + b_y\right), \tag{7}$$

where h_t and h'_t represent the hidden states output by the forward LSTM and the backward LSTM, respectively. W_y and W'_y denote the weight matrices, and b_y is the bias term.

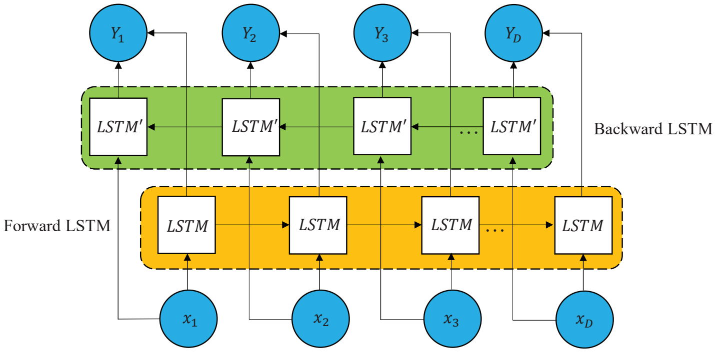

Figure 2. Bi-LSTM module structure

The LSTM structure, taking the forward LSTM as an example, is shown in Figure 3, where i_t, o_t, and f_t represent the input gate, output gate, and forget gate, respectively. $\tilde{C}_t$ denotes the temporary cell state, while C_t represents the cell state. The hidden state is denoted as h_t, and x_t is the feature matrix representing the user's check-in behavior at time step t. W_i, W_f, W_c, and W_o are the weight matrices for the input gate, forget gate, temporary cell state, and output gate, respectively; these weight matrices are parameters to be learned. Specifically, the input gate i_t controls the input of the user's check-in behavior features, the output gate o_t regulates the output of these features, and the forget gate f_t manages retention or forgetting of the user's check-in behavior features, allowing the model to learn the dynamic changes in user check-in behaviors through these three gates.

The forward LSTM calculations are shown in Equations (8), (9), (10), (11), (12), and (13).

$$i_t = \sigma\left(W_{it}x_t + W_{ih}h_{t-1} + b_i\right), \tag{8}$$

$$f_t = \sigma\left(W_{ft}x_t + W_{fh}h_{t-1} + b_f\right), \tag{9}$$

$$\tilde{C}_t = tanh\left(W_{ct}x_t + W_{ch}h_{t-1} + b_c\right), \tag{10}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, \tag{11}$$

$$o_t = \sigma\left(W_{ot}x_t + W_{oh}h_{t-1} + b_o\right), \tag{12}$$

$$h_t = o_t \odot tanh\left(C_t\right), \tag{13}$$

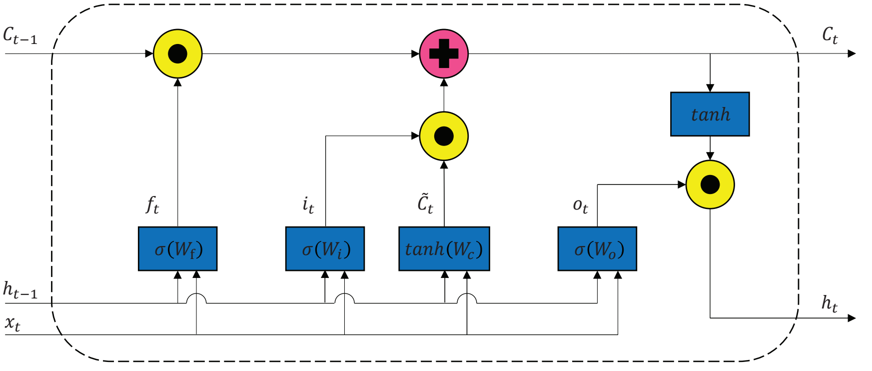

Figure 3. The Forward LSTM structure

where b_i, b_f, b_c, and b_o represent the bias terms for the input gate, forget gate, temporary cell state, and output gate, respectively. These bias terms are parameters that need to be learned. σ is the sigmoid activation function, and $\odot$ denotes element-wise multiplication. The structure and notation of the backward LSTM are similar to those of the forward LSTM and will not be repeated.

3.2.3 Attention Mechanism Module

The gate structure of the Bi-LSTM module selectively saves and forgets user check-in features. When dealing with long sequences, it is prone to issues such as forgetting key information or information redundancy. Additionally, since the Bi-LSTM module performs two computations at each time step, the model is more susceptible to problems like gradient explosion or gradient vanishing, which can lead to a decrease in model accuracy.

The ProbSparse attention mechanism [18] can dynamically allocate weights to input features, identifying important information and irrelevant data, thereby alleviating the problems of key information loss and irrelevant information redundancy. Furthermore, compared to traditional attention mechanisms, the ProbSparse attention mechanism is more efficient in filtering information, reducing the amount of similarity calculations, and mitigating issues related to gradient explosion or gradient vanishing, improving the operational efficiency of the model.

Therefore, the ProbSparse attention mechanism is introduced to calculate weight information for different positions based on the importance of various segments of the user check-in sequence. Accesses with higher weight are given greater attention, focusing on essential information. The input to the attention mechanism, represented as Q, K, and V, denotes the query vector, key values, and similarity weights, respectively, with the calculations shown in Equation (14).

$$
\begin{aligned}
Q &= W_Q Y, \\
K &= W_K Y, \\
V &= W_V Y,
\end{aligned}
\tag{14}
$$

where $Y = [Y_1, Y_2, \ldots, Y_D]$ represents the final output of the Bi-LSTM module, W_Q, W_K, and W_V are the weight parameters.

Utilizing the dot product of Q and K to assign weights to input features is a core step in the attention mechanism. The dot product reflects the strength of the correlation between the user's current preferences and the candidate interest points. Traditional attention mechanisms require considering all possible dot products for computation; however, in practice, only a small number of these dot products contribute significantly to the model. Therefore, the traditional attention mechanism involves a significant amount of redundant computations, which leads to a decrease in model accuracy and lower efficiency.

The ProbSparse attention mechanism addresses this issue by assessing the contribution size of different dot products during computation. It assigns computational weights to these dot products, ignoring those with relatively low contributions, thereby reducing the amount of similarity calculations needed and alleviating gradient issues. Specifically, the ProbSparse attention mechanism uses the Kullback-Leibler (KL) divergence to measure the difference between the attention probability distribution of the vector Q corresponding to the dot products and a uniform distribution. It ignores the dot products where the difference is minimal. The distribution equations are shown in Equations (15), (16), and (17).

$$
k(q_i, k_j) = \exp\left(\frac{q_i k_j^T}{\sqrt{d}}\right), \tag{15}
$$

$$
p(k_j \mid q_i) = \frac{k(q_i, k_j)}{\sum_l k(q_i, k_l)}, \tag{16}
$$

$$
q(k_j \mid q_i) = \frac{1}{L_k}, \tag{17}
$$

where $p(k_j \mid q_i)$ and $q(k_j \mid q_i)$ represent the probability distribution of the attention mechanism and the uniform distribution, respectively. $\exp\left(\frac{q_i k_j^T}{\sqrt{d}}\right)$ is the asymmetric exponential kernel function, L_k denotes the length of vector Q. q_i and k_j refer to the i^{th} row of vectors Q and K, respectively. The sparsity measurement equation is then obtained using KL divergence, as shown in Equation (18).

$$
M(q_i, K) = ln \sum_{j=1}^{L_K} e^{\frac{q_i k_j^T}{\sqrt{d}}} - \frac{1}{L_K} \sum_{j=1}^{L_K} \frac{q_i k_j^T}{\sqrt{d}}. \tag{18}
$$

If the value of $M(q_i, K)$ corresponding to a certain vector Q_i is large, it indicates that Q_i contains dot products with greater contributions. The top $n = c \ln L_Q$ vectors of Q with the largest values of M are selected for computation, where L_Q is the temporal length of vector Q and c is a constant. The calculation for the ProbSparse attention mechanism is shown in Equation (19).

$$A(Q, K, V) = softmax\left(\frac{\bar{Q}K^T}{\sqrt{d}}\right)V, \tag{19}$$

where $\bar{Q}$ is the sparse matrix formed by the top n vectors of Q with the largest values of M, and $A(Q, K, V)$ represents the output of the ProbSparse attention mechanism.

3.2.4 POI Matching Module

The POI matching module includes a fully connected layer and a Softmax layer. The outputs from the Bi-LSTM module and the attention mechanism module are fed into the POI matching module, which then outputs the model's predicted probability distribution through the Softmax layer, as shown in Equation (20).

$$\tilde{y} = softmax\left(W_y A(Q, K, V)^T + F(T) + F(L)\right), \tag{20}$$

where $A(Q, K, V)$ represents the output of the attention mechanism module, $F(T)$ denotes the time interval matrix, $F(L)$ indicates the spatial interval matrix, and W_y is a learnable parameter matrix.

3.2.5 Loss Function

The cross-entropy loss function is utilized to optimize the model's learning performance. N_x represents the negative samples, while N_y represents the positive samples. Given the user check-in sequence $N(u_i)$, $N_x = [P^i_{seq_{D+1}}, P^i_{seq_{D+2}}, \ldots, P^i_{seq_{D+m}}]$ represents the randomly selected locations that user u_i has not visited in the prediction results, and N_y denotes the correct predicted locations. The calculation for the cross-entropy loss function is shown in Equation (21).

$$Loss = -\sum_i \sum_D \left(\log \sigma(N_y) + \sum_{x \neq y} \log(1 - \sigma(N_x))\right). \tag{21}$$

3.2.6 Algorithm Design

In the PMBAM model that integrates Bi-LSTM and attention mechanism, the original data processing and training process is illustrated in Algorithm 1.

User check-in data is first sorted in chronological order. Subsequently, features for time intervals, spatial intervals, and location encoding are established based on the time and positional encoding in the dataset. The user check-in feature matrix, which contains various contextual information, is then input into the model for training. The training results are obtained, and the loss function is computed; the Adam optimizer is used to update the model parameters, optimizing the model training. Finally, various metrics are used to evaluate the performance of the algorithm.

Algorithm 1 PMBAM Algorithm

Input: POI dataset
Output: The user's next POI recommendation

```
// Original Data Processing
for user in (1, user-number):
    check-in order by time
    N(u_i) = [P^i_seq1, P^i_seq2, P^i_seq3, ..., P^i_seqD]
    △^t_ij ← Equation (1)
    △^d_ij ← Equation (2)
    t_ss
    PE_(p_i,2f) ← Equation (3)
    PE_(p_i,2f+1) ← Equation (4)
end
// Model Training
for x in (1, epoch-number):
    for y in (1, batch-number):
        Model (N(u_i), △^t_ij, △^d_ij, t_ss, PE_(p_i,2f), PE_(p_i,2f+1))
        select POI from all locations
        Loss ← Equation (21)
        Recall@K = (POI, Label) ← Equation (22)
        NDCG@K = (POI, Label) ← Equation (23)
    end
end
```

4 EXPERIMENT AND RESULTS ANALYSIS

4.1 Experimental Setup

4.1.1 Dataset

The experiments are conducted using the Gowalla dataset and the Foursquare-NYC dataset. The Gowalla dataset includes check-in data shared by users on the Gowalla site from February 2009 to October 2010 in California and Nevada, USA. The Foursquare-NYC dataset is sourced from the Foursquare website and contains user check-in records from April 2012 to February 2013 in New York, USA. Each dataset uses 80 % of the data as the training set, 10 % as the validation set, and the remaining 10 % as the test set. To reduce the negative impact of abnormal data on the experimental results, users with fewer than 10 total check-ins and locations with fewer than 10 total check-ins are removed from both datasets. Only users and check-in locations with 10 or more interaction behaviors are retained. The details of the cleaned dataset are shown in Table 1.

	Gowalla	Foursquare-NYC
Users	29 858	1 083
Locations	40 981	42 981
Check-ins	1 027 370	227 428

Table 1. Cleaned dataset details

4.1.2 Evaluation Metrics

Model performance is evaluated using mainstream metrics, specifically $Recall@K$ and Normalized Discounted Cumulative Gain $(NDCG)@K$, with (K) set to 5 and 10.

$Recall@K$ can measure the overlap between the top-K recommendation results and the user's actual check-in locations, which refers to the proportion of the user's actual check-in locations present in the model's predicted results, as shown in Equation (22).

$$Recall@K = \frac{|R \cap R_{1:\hat{K}}|}{|R|}, \tag{22}$$

where $|R|$ represents the set of the user's actual check-in locations, and $|R \cap R_{1:\hat{K}}|$ represents the top-K recommended results generated by the model.

$NDCG@K$ can measure the ranking quality of the predicted results. The closer the correctly predicted results are to the top of the top-K recommendation results, the larger the value of $NDCG@K$, as shown in Equation (23).

$$NDCG@K = \frac{DCG_K}{IDCG_K}, \tag{23}$$

where DCG_K represents the actual ranking score of the correctly predicted results in the top-K recommendation results, while $IDCG_K$ represents the ranking score when the correctly predicted results are ranked first among the top-K recommendation results, which is the ideal ranking score. The calculations for DCG_K and $IDCG_K$ are shown in Equations (24) and (25).

$$DCG_K = \sum_{j=1}^{K} \frac{2^{rel(j)} - 1}{\log_2(j+1)}, \tag{24}$$

$$IDCG_K = \sum_{j=1}^{|REL|} \frac{2^{rel(j)} - 1}{\log_2(j+1)}, \tag{25}$$

where $rel(j)$ represents the relevance score of the predicted point of interest at position j. If the point of interest is present at position j, then $rel(j) = 1$; if the point of interest is not present at position j, then $rel(j) = 0$. $|REL|$ denotes the set of the top-K results from the predictions arranged in descending order of their relevance scores.

4.1.3 Parameter Settings

The learning rate is set to 0.003, the number of attention heads in the ProbSparse attention mechanism is set to 4, the batch size is set to 1 024, the number of negative samples is set to 10, the dropout rate is set to 0.2, and the Adam optimizer is employed for model training.

4.2 Experimental Analysis

4.2.1 Comparative Experiments

The experiments are conducted based on the open-source model framework of LibCity [19]. LibCity is a standardized traffic prediction framework that includes various mainstream traffic prediction models and spatiotemporal datasets, utilizing a unified data format and model interface. The proposed PMBAM model is compared against six mainstream models: FPMC [20], ST-RNN [21], DeepMove [22], LSTPM [23], GeoSAN [24], as well as four advanced models: STAN [25], EEDN [10], STSASP [26] and STHGCN [12]. Detailed information about the baseline methods is provided below:

FPMC [20]: This method combines personalized transition graphs of Markov chains with matrix factorization to learn user-specific preferences, while also introducing a Bayesian personalized ranking optimization model for parameter adjustment. It serves as a classic baseline method.

ST-RNN [21]: This method incorporates temporal information and spatial information into a recurrent neural network (RNN) model to effectively model spatiotemporal features.

DeepMove [22]: This method is based on attention-based recurrent networks, capturing user preferences through multimodal embedding recurrent neural networks, and utilizes historical attention models to capture the periodicity of user preferences.

LSTPM [23]: This method simultaneously models users' long-term and short-term preferences to generate POI recommendation results.

GeoSAN [24]: This method proposes a new loss function that integrates a geography encoder based on self-attention mechanisms with a geography-aware negative sampler to model geographic information.

STAN [25]: This method takes into account the importance of non-continuous information in user check-in records, employing a dual-layer attention mechanism architecture along with the relative spatiotemporal information of user check-ins to facilitate POI recommendations.

EEDN [10]: This method is based on graph structures, integrating graph convolutional networks into the encoder-decoder framework to alleviate the cold start problem of check-in datasets.

STSASP [26]: This method utilizes various contextual information, integrating gated recurrent unit networks and self-attention mechanisms to recommend sequences of points of interest to users.

STHGCN [12]: This method leverages collaborative information between an individual user's trajectory and those of other users to learn user preferences, while also incorporating a hypergraph transformer that combines hypergraph structural encoding with contextual information.

The proposed POI recommendation model that integrates an attention mechanism is compared with other models on the Gowalla and Foursquare-NYC datasets. The evaluation metrics used are $Recall@5$, $Recall@10$, $NDCG@5$, and $NDCG@10$. The comparison results are presented in Tables 2 and 3, and the improvements of PMBAM over the optimal baseline model STHGCN are highlighted in the tables.

	Recall@5	*Recall@10*	*NDCG@5*	*NDCG@10*
FPMC(2010)	0.1824	0.2473	0.1611	0.1762
ST-RNN(2016)	0.2036	0.2589	0.1703	0.1828
STGN(2018)	0.2093	0.2802	0.1986	0.2195
DeepMove(2018)	0.2404	0.3225	0.2365	0.2513
LSTPM(2020)	0.2250	0.3027	0.2489	0.2711
GEOSAN(2020)	0.2912	0.3980	0.2372	0.2865
STAN(2021)	0.3381	0.4296	0.2545	0.3160
EEDN(2023)	0.3477	0.4383	0.2589	0.3241
STSASP(2023)	0.3745	0.4561	0.2693	0.3332
STHGCN(2023)	0.3791	0.4558	0.2734	0.3378
PMBAM	**0.4229**	**0.4922**	**0.3119**	**0.3523**
Improvements	11.55 %	7.98 %	14.08 %	4.29 %

Table 2. Comparison results of different models on Foursquare

The experimental results indicate that FPMC performs the worst across both datasets. This is attributed to the fact that traditional Markov chain-based POI recommendation models are inferior in performance compared to deep learning models. Through the application of deep learning, the model can better handle the nonlinear relationships inherent in check-in sequence, and exhibit enhanced flexibility and generalization capability.

In the context of deep learning baseline methods, models based on attention mechanisms (GEOSAN, STAN, STHGCN, STSASP) exhibit superior recommendation capabilities in most cases compared to recurrent neural network-based models (ST-RNN, STGN, DeepMove, LSTPM). This enhancement is primarily attributed

	Recall@5	*Recall*@10	$NDCG$@5	$NDCG$@10
FPMC(2010)	0.0923	0.1267	0.0473	0.0523
ST-RNN(2016)	0.1262	0.1674	0.0643	0.0766
STGN(2018)	0.1211	0.1625	0.0639	0.0761
DeepMove(2018)	0.1443	0.1851	0.0792	0.0903
LSTPM(2020)	0.1517	0.1930	0.0875	0.1022
GEOSAN(2020)	0.2225	0.2585	0.1627	0.1799
STAN(2021)	0.2571	0.2783	0.1744	0.1905
EEDN(2023)	0.2594	0.2867	0.1785	0.1961
STSASP(2023)	0.2847	0.3256	0.1979	0.2104
STHGCN(2023)	0.2883	0.3306	0.1998	0.2145
PMBAM	**0.3108**	**0.3437**	**0.2177**	**0.2281**
Improvements	7.80 %	3.96 %	8.95 %	6.34 %

Table 3. Comparison results of different models on Gowalla

to the attention mechanism's ability to effectively filter important data within the dataset, thereby demonstrating a stronger capacity for capturing dependencies.

The proposed PMBAM model significantly outperforms other baseline methods in terms of *Recall*@K and *NDCG*@K. Compared to the best baseline method, STHGCN, which also employs attention mechanism, the PMBAM model achieves an improvement of 9.60 % and 8.67 % in *Recall*@K and *NDCG*@K, respectively, on the Foursquare dataset (averaging these two results as a common practice). On the Gowalla dataset, the improvements are 5.75 % and 7.60 % for *Recall*@K and *NDCG*@K, respectively. The STHGCN model captures the collaborative information between a single user's trajectory and those of other users using a graph structure, achieving commendable results in the next point of interest recommendation task. However, the graph structure fails to learn the sequential dependencies of user check-in behaviors, which affects the overall recommendation performance of the model. We employ Bi-LSTM to comprehensively consider both forward and backward contextual information, establishing connections between a user's current check-in location and both historical and future check-in locations. Additionally, ProbSparse attention mechanism is utilized to alleviate issues of gradient explosion and vanishing gradients to a certain extent, resulting in improved recommendation performance.

In summary, the proposed PMBAM model demonstrates superior recommendation performance compared to baseline methods, while also validating the effectiveness of the attention mechanism in point-of-interest recommendations. Furthermore, the PMBAM model shows improvements across various metrics on both the Foursquare and Gowalla datasets, indicating its robust performance when dealing with both limited and abundant points of interest.

4.2.2 Ablation Experiments

By applying various transformations to the PMBAM model, we aim to demonstrate the effectiveness of each module within the model. The specific variants of the PMBAM model are as follows:

PMBAM-ST: In the Embedding module, only the check-in record sequence and positional encoding features are considered, while ignoring the spatial-temporal interval information.

PMBAM-BI: In the Bi-LSTM module, the bidirectional LSTM, which processes the check-in sequence in both forward and backward directions, is replaced with a unidirectional LSTM.

PMBAM-AT: In the attention mechanism module, the ProbSparse attention mechanism is replaced with a traditional self-attention mechanism.

	Recall@5	*Recall@10*	*NDCG@5*	*NDCG@10*
PMBAM	0.4229	0.4922	0.3119	0.3523
PMBAM-ST	0.3272	0.4051	0.2516	0.3008
PMBAM-BI	0.3685	0.4368	0.2691	0.3202
PMBAM-AT	0.3844	0.4605	0.2873	0.3385

Table 4. Comparison results of the ablation experiments on Foursquare

	Recall@5	*Recall@10*	*NDCG@5*	*NDCG@10*
PMBAM	0.3105	0.3437	0.2177	0.2281
PMBAM-ST	0.2463	0.2874	0.1715	0.1943
PMBAM-BI	0.2817	0.3098	0.1986	0.2142
PMBAM-AT	0.2952	0.3213	0.2054	0.2201

Table 5. Comparison results of the ablation experiments on Gowalla

The experimental results are illustrated in Tables 4 and 5. The PMBAM-ST model, due to the exclusion of spatial-temporal interval information from the user's check-in records, fails to effectively capture the dynamic changes in user preference behavior. This deficiency results in a decrease in *Recall@K* and *NDCG@K* by 24.96 % and 20.23 %, respectively, on the Foursquare dataset, and by 22.63 % and 21.86 % on the Gowalla dataset.

Similarly, the PMBAM-BI model suffers performance degradation as it does not effectively integrate bidirectional contextual information. On the Foursquare dataset, *Recall@K* and *NDCG@K* decrease by 13.63 % and 12.71 %, respectively, while on the Gowalla dataset, they decline by 10.65 % and 7.99 %.

Furthermore, the PMBAM-AT model experiences a reduction in performance due to the replacement of the probabilistic sparse attention mechanism with a traditional self-attention mechanism. This leads to a decrease in *Recall@K* and

NDCG@K by 8.30 % and 6.13 %, respectively, on the Foursquare dataset, and by 6.16 % and 4.77 % on the Gowalla dataset.

Through comparative analysis, it is evident that the PMBAM model significantly outperforms the PMBAM-ST, PMBAM-BI, and PMBAM-AT models in both *Recall@K* and *NDCG@K* metrics. This validates the positive contributions of the proposed modules to enhancing the performance of the POI recommendation algorithm. Additionally, it underscores the importance of deeply considering the complex dynamic changes in user check-in behaviors, which have a substantial impact on the effectiveness of POI recommendation algorithms.

4.2.3 Hyperparameter Analysis

The dimension of the embedding vector d has a certain impact on the recommendation performance of the model. A dimension that is too small may lead to information loss, while a dimension that is too large can result in overfitting during model training. Therefore, both excessively large and excessively small embedding dimensions can cause a decline in model accuracy.

We test the effect of varying the embedding vector dimension d within the range of 10 to 60, using a step size of 10, while keeping other hyperparameters constant. The evaluation metric used is *Recall@5*. The testing results are illustrated in Table 6, which shows that the evaluation metric reaches its peak when the embedding vector dimension is set to 50.

	Foursquare	Gowalla
10	0.4144	0.2916
20	0.4159	0.2985
30	0.4180	0.3028
40	0.4193	0.3060
50	0.4229	0.3108
60	0.4213	0.3096

Table 6. Performance of different embedding vector dimensions on two datasets

Selecting an appropriate dropout rate can effectively alleviate the overfitting problem that occurs during the training of deep learning models. Under the premise of keeping other hyperparameters unchanged, we tested the impact of different dropout rates on the model's recommendation performance, with *Recall@5* as the evaluation metric. As shown in Table 7, when the dropout rate is set to 0.2, the proposed PMBAM model achieves better experimental results.

5 CONCLUSIONS

In this paper, we propose a POI recommendation model named PMBAM that integrates an attention mechanism to deeply explore the contextual information of

	Foursquare	Gowalla
0.0	0.2071	0.1628
0.2	0.4229	0.3104
0.4	0.3915	0.2902
0.6	0.4083	0.2994

Table 7. Performance of different dropout on two datasets

user check-in sequence and capture the complex dynamic changes in user preference behaviors. Firstly, the use of Bi-LSTM allows for processing data in both forward and backward directions, enabling the integration of various contextual information to learn the dynamic variations in the user check-in behavior. Secondly, the probabilistic sparse attention mechanism efficiently filters information while alleviating the issues of key information loss or redundancy that can occur when Bi-LSTM handles long sequences.

Experimental results on two real-world datasets demonstrate that the PM-BAM model outperforms previously proposed advanced models in both *Recall@K* and *NDCG@K* metrics, proving the effectiveness and superiority of the proposed attention-based POI recommendation model. Furthermore, ablation studies validate the rationality of different modules within the model.

In future work, we will continue to optimize the model structure for the POI recommendation problem and further uncover hidden information in user check-ins data. Additionally, we aim to explore how the complex dynamic changes in the user check-in behavior impacts POI recommendations, thereby enhancing the accuracy of predicting user preference behaviors.

6 DECLARATIONS

The funding for this research is as follows:

1. National key research and development programme (No. 2022YFD2000500);
2. National Natural Science Foundation of China (No. 62071157);
3. National Science Foundation of Heilongjiang Province (No. LH2022F034).

REFERENCES

[1] Liu, Y.—Pei, A.—Wang, F.—Yang, Y.—Zhao, X.—Wang, H.—Dai, H.—Qi, L.—Ma, R.: An Attention-Based Category-Aware GRU Model for the Next POI Recommendation. International Journal of Intelligent Systems, Vol. 36, 2021, No. 7, pp. 3174–3189, doi: 10.1002/int.22412.

[2] Shi, M.—Shen, D.—Kou, Y.—Nie, T.—Yu, G.: Next Point-of-Interest Recommendation Approach with Global and Local Feature Fusion. Journal of Software, Vol. 34, 2023, No. 2, pp. 786–801, doi: 10.13328/j.cnki.jos.006712 (in Chinese).

[3] Tang, J.—Chen, Y.—Zhou, M.—Wang, X.: A Survey of Studies on Deep Learning Applications in POI Recommendation. Computer Engineering, Vol. 48, 2022, No. 1, pp. 12–23, doi: 10.19678/j.issn.1000-3428.0061598 (in Chinese).

[4] Qin, Y.—Wang, Y.—Sun, F.—Ju, W.—Hou, X.—Wang, Z.—Cheng, J.—Lei, J.—Zhang, M.: DisenPOI: Disentangling Sequential and Geographical Influence for Point-of-Interest Recommendation. Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23), 2023, pp. 508–516, doi: 10.1145/3539597.3570408.

[5] Meng, L.—Liu, Z.—Chu, D.—Sheng, Q. Z.—Yu, J.—Song, X.: POI Recommendation for Occasional Groups Based on Hybrid Graph Neural Networks. Expert Systems with Applications, Vol. 237, Part B, 2024, Art. No. 121583, doi: 10.1016/j.eswa.2023.121583.

[6] Wang, Y.—Ma, Y.—Xu, S.—Wang, Y.—Luo, A.—Liu, W.—Di, L.: Research Progress and Prospects of Point of Interest Recommendation Method. Surveying and Mapping Science, Vol. 48, 2023, No. 12, pp. 217–224, doi: 10.16251/j.cnki.1009-2307.2023.12.021 (in Chinese).

[7] Zhang, J.—Liu, X.—Zhou, X.—Chu, X.: Leveraging Graph Neural Networks for Point-of-Interest Recommendations. Neurocomputing, Vol. 462, 2021, pp. 1–13, doi: 10.1016/j.neucom.2021.07.063.

[8] Hu, H. X.—Jiang, Z. W.—Zhao, Y. F.—Zhang, Y.—Wang, H.—Wang, W.: Network Representation Learning-Enhanced Multisource Information Fusion Model for POI Recommendation in Smart City. IEEE Internet of Things Journal, Vol. 8, 2021, No. 12, pp. 9539–9548, doi: 10.1109/JIOT.2020.3006989.

[9] Xu, C.—Ding, A. S.—Zhao, K.: A Novel POI Recommendation Method Based on Trust Relationship and Spatial-Temporal Factors. Electronic Commerce Research and Applications, Vol. 48, 2021, Art. No. 101060, doi: 10.1016/j.elerap.2021.101060.

[10] Wang, X.—Fukumoto, F.—Cui, J.—Suzuki, Y.—Li, J.—Yu, D.: EEDN: Enhanced Encoder-Decoder Network with Local and Global Context Learning for POI Recommendation. Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), 2023, pp. 393–392, doi: 10.1145/3539618.3591678.

[11] Lan, P.—Zhang, Y.—Xiang, H.—Wang, Y.—Zhou, W.: Spatio-Temporal Position-Extended and Gated-Deep Network for Next POI Recommendation. In: Wang, X., Sapino, M. L., Han, W. S. et al. (Eds.): Database Systems for Advanced Applications (DASFAA 2023). Springer, Cham, Lecture Notes in Computer Science, Vol. 13944, 2023, pp. 505–520, doi: 10.1007/978-3-031-30672-3_34.

[12] Yan, X.—Song, T.—Jiao, Y.—He, J.—Wang, J.—Li, R.—Chu, W.: Spatio-Temporal Hypergraph Learning for Next POI Recommendation. Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), 2023, pp. 403–412, doi: 10.1145/3539618.3591770.

[13] Yang, S.—Liu, J.—Zhao, K.: GETNext: Trajectory Flow Map Enhanced Transformer for Next POI Recommendation. Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), 2022, pp. 1144–1153, doi: 10.1145/3477495.3531983.

[14] Li, Q.—Xu, X.—Liu, X.—Chen, Q.: An Attention-Based Spatiotemporal GGNN for Next POI Recommendation. IEEE Access, Vol. 10, 2022, pp. 26471–26480, doi: 10.1109/ACCESS.2022.3156618.

[15] Liu, Y.—Song, Z.—Xu, X.—Rafique, W.—Zhang, X.—Shen, J.—Khosravi, M. R.—Qi, L.: Bidirectional GRU Networks-Based Next POI Category Prediction for Healthcare. International Journal of Intelligent Systems, Vol. 37, 2022, No. 7, pp. 4020–4040, doi: 10.1002/int.22710.

[16] Kasgari, A. B.—Safavi, S.—Nouri, M.—Hou, J.—Sarshar, N. T.—Ranjbarzadeh, R.: Point-of-Interest Preference Model Using an Attention Mechanism in a Convolutional Neural Network. Bioengineering, Vol. 10, 2023, No. 4, Art. No. 495, doi: 10.3390/bioengineering10040495.

[17] Sun, X.—Huang, B.—Wang, X.—Yu, D.: MARAN: Supporting Awareness of Users' Routines and Preferences for Next POI Recommendation Based on Spatial Aggregation. Expert Systems with Applications, Vol. 238, Part F, 2024, Art. No. 121961, doi: 10.1016/j.eswa.2023.121961.

[18] Zhou, H.—Zhang, S.—Peng, J.—Zhang, S.—Li, J.—Xiong, H.—Zhang, W.: Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, No. 12, pp. 11106–11115, doi: 10.1609/aaai.v35i12.17325.

[19] Wang, J.—Jiang, J.—Jiang, W.—Li, C.—Zhao, W. X.: LibCity: An Open Library for Traffic Prediction. Proceedings of the 29th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '21), ACM, 2021, pp. 145–148, doi: 10.1145/3474717.3483923.

[20] Rendle, S.—Freudenthaler, C.—Schmidt-Thieme, L.: Factorizing Personalized Markov Chains for Next-Basket Recommendation. Proceedings of the 19th International Conference on World Wide Web (WWW '10), ACM, 2010, pp. 811–820, doi: 10.1145/1772690.1772773.

[21] Liu, Q.—Wu, S.—Wang, L.—Tan, T.: Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, 2016, No. 1, pp. 194–200, doi: 10.1609/aaai.v30i1.9971.

[22] Feng, J.—Li, Y.—Zhang, C.—Sun, F.—Meng, F.—Guo, A.—Jin, D.: DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. Proceedings of the 2018 World Wide Web Conference (WWW '18), ACM, 2018, pp. 1459–1468, doi: 10.1145/3178876.3186058.

[23] Sun, K.—Qian, T.—Chen, T.—Liang, Y.—Nguyen, Q. V. H.—Yin, H.: Where to Go Next: Modeling Long- and Short-Term User Preferences for Point-of-Interest Recommendation. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, No. 1, pp. 214–221, doi: 10.1609/aaai.v34i01.5353.

[24] Lian, D.—Wu, Y.—Ge, Y.—Xie, X.—Chen, E.: Geography-Aware Sequential Location Recommendation. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '20), 2020, pp. 2009–2019, doi: 10.1145/3394486.3403252.

[25] Luo, Y.—Liu, Q.—Liu, Z.: STAN: Spatio-Temporal Attention Network for Next

Location Recommendation. Proceedings of the Web Conference 2021 (WWW '21), ACM, 2021, pp. 2177–2185, doi: 10.1145/3442381.3449998.

[26] Zhu, J.—Ma, W.—Wang, B.—Wu, C.: Sequential Recommendation of Point of Interest Combining Spatio-Temporal Network with Self-Attention. Computer Engineering and Design, Vol. 44, 2023, No. 2, pp. 590–597, doi: 10.16208/j.issn1000-7024.2023.02.037 (in Chinese).

Yuan Cheng works as Associate Professor with the School of Computer Science and Technology, Harbin University of Science and Technology. She received her Ph.D. degree from the Harbin Engineering University. Her research interests include data mining, uncertainty analysis, and spatiotemporal data analysis.

Yu Cheng is a third-year graduate student at the School of Computer Science and Technology, Harbin University of Science and Technology. His main research interest is point-of-interest recommendation systems.

Ao Li is Full Professor at the School of Computer Science and Technology, Harbin University of Science and Technology. He received his Ph.D. degree from the Harbin Engineering University. His research interests include machine learning, pattern recognition and computer vision.

Feihu Jin works as Associate Professor with the School of Computer Science and Technology, Harbin University of Science and Technology. He received his Ph.D. degree from the Harbin Institute of Technology. His research interests include artificial intelligence and intelligent control.