Interaction-enhanced and Time-aware Graph Convolutional Network for Successive Point-of-Interest Recommendation in Travelling Enterprises

Yuwen Liu, Huiping Wu, Khosro Rezaee, Mohammad R. Khosravi, Osamah Ibrahim Khalaf, Arif Ali Khan, Dharavath Ramesh, and Lianyong Qi*

***Abstract*— Extensive user check-in data incorporating user preferences for location is collected through Internet of Things (IoT) devices. including cell phones and other sensing devices in Location-based Social Network (LBSN). It can help travelling enterprises intelligently predict users' interests and preferences, provide them with scientific tourism paths and increase the enterprises income. Thus, successive Point-of-Interest (POI) recommendation has become hot research topic in Augmented Intelligence of Things (AIoT). Presently, various methods have been applied to successive POI recommendations. Among them, the Recurrent Neural Network (RNN)-based approaches are committed to mining the sequence relationship between POIs, but ignore the high-order relationship between users and POIs. The Graph Neural Network (GNN)-based methods can capture the high-order connectivity, but it does not take the dynamic timeliness of POIs into account. Therefore, we propose an Interaction-enhanced and Time-aware Graph Convolution Network (ITGCN) for successive POI recommendation. Specifically, we design an improved graph convolution network for learning the dynamic representation of users and POIs. We also designed a self-attention aggregator to embed high-order connectivity into the node representation selectively. The Enterprise Management Systems (EMS) can predict the preferences of users, which is helpful for future planning and development. Finally, experimental results prove that ITGCN bring better results compared to the existing methods.**

***Index Terms*— Augmented Intelligence of Things, aggregator, Graph Convolution Network, self-attention, successive Point-of-Interest recommendation**

Y. Liu and L. Qi [Corresponding author] are with College of Computer Science and Technology, China University of Petroleum, Qingdao 266580, China (email: yuwenliu97@gmail.com; lianyongqi@gmail.com).

H. Wu is with Shandong Provincial University Laboratory for Protected Horticulture, Weifang University of Science and Technology, China (email: wuhp760731@wfust.edu.cn).

K. Rezaee is with Meybod University, Meybod, Iran (email: kh.rezaee@meybod.ac.ir).

M. R. Khosravi is with is with Shandong Provincial University Laboratory for Protected Horticulture, Weifang University of Science and Technology, Weifang 262799, Shandong, China, and Department of Computer Engineering, Persian Gulf University, Bushehr 75169, Iran (e-mail: m.khosravi@wfust.edu.cn).

O. I. Khalaf is with Al-Nahrain Nanorenewable Energy Research Center, Al-Nahrain University, Iraq (email: usama81818@nahrainuniv.edu.iq).

A. A. Khan is with M3S Empirical Software Engineering Research Unit, University of Oulu, Oulu, Finland (email:arif.khan@oulu.fi).

D. Ramesh is with Department of Computer Science and Engineering, Indian Institute of Technology (ISM), India (email:drramesh@iitism.ac.in).

I. INTRODUCTION

WITH the rapid development of sensing technology [1], Augmented Intelligence of Things (AIoT) [2] [3] and smart city construction [4] increase the data amount exponentially. To help users mine relevant information from the massive data, the recommendation system came into being. The recommendation systems provided great support for the Enterprise Management Systems (EMS) [5]. The smart enterprise systems collect the behavior data of users through sensors, so that the smart enterprise can use the historical data to predict the users' preferences. After mastering the users' preferences by using appropriate recommendation algorithms, smart enterprises can carry out targeted enterprise planning and work arrangements, so as to improve economic benefits and promote the construction of the smart city. Specifically, the successive Point-of-Interest (POI) recommendation has dramatically promoted the intelligent development of travelling enterprises. Through the historical interest preferences of users, travelling enterprises can predict the POIs that users may like in the future to make intelligent recommendations. This method can significantly improve user satisfaction and enterprise income. Using users' check-in to predict users' POI preference is also important research in computer science, sociology, and complexity science. It also has important practical and academic value in location-based social network (LBSN) services, smart city construction, augmented Intelligence of Things for smart enterprise systems, etc.

In LBSNs, people can use their location sensors to check in at their favorite POIs. Therefore, travelling enterprises can mine users' interests from check-in data to predict the POIs that users may like in the future. However, the traditional POI recommendation often faces a problem: it ignores the timeliness of users' check-in. It only predicts the possible check-ins of users in the future, ignoring the specific time information. Suppose we predict the POIs that users may be interested in the next six months, which often has low practical value. Therefore, to make the POI recommendation more timely, successive POI recommendation has attracted more researchers' attention. This research needs more focus on the sequence information of users' check-in.

Presently, most researchers focus on the one-way information of the sequence, and use machine learning [6] or deep learning [7] to make successive POI recommendations, such as

Hidden Markov Model (HMM) or Recurrent Neural Network (RNN). However, the HMM-based method has no after effect. It assumes that the user's next location is only related to the current location, but not to other historical check-ins. Obviously, this assumption does not correspond to the actual situation. The users' historical check-ins often have different influences on future check-ins. Thus, the HMM-based methods often perform poorly in long sequence recommendations. Although the RNN-based method has been able to mine the long sequence information, it still faces some challenges. First, it focuses on users and ignores the temporal dynamics of POIs. In fact, POIs have stable static attributes and unstable dynamic attributes. A POI may show different temporal dynamics at different time points [8]. Second, although the RNN can process the users' sequence information well, it only considers the first-order interaction. And it ignores the high-order connectivity between users and POIs [9].

To solve the above challenges, we proposed an **I**nteraction-enhanced and **T**ime-aware **G**raph **C**onvolutional **N**etwork (ITGCN) for successive POI recommendation in travelling enterprises. Firstly, we construct a bipartite graph of the interaction between POIs and users, and mine their hidden features. Then, we use GCN to get the embedding representation of users and POIs. In addition, to model the sequence information of users and POIs at the same time and realize its dynamic representation, we combine GCN with the self-attention aggregator. Through the above design, we can capture the sequence information and the high-order connectivity of user interaction with POIs. In short, our contributions are summarized as follows:

- We propose an Interaction-enhanced and Time-aware GCN for successive POI recommendations in travelling enterprises. It can learn the indirect relationship between users and POIs, and realize the dynamic representation of the nodes.
- We design a novel aggregator to embed high-order connectivity into the node representation selectively. At the same time, the aggregator can pay attention to the influence of interaction location and time information. It can also be used to update the dynamic representation of nodes.
- We have conducted experiments on four datasets. The experiments prove that ITGCN brings better results compared to the existing methods.

II. Related Work

A. Traditional POI recommendation

The traditional POI recommendation in travelling enterprises mainly focuses on the historical POIs of users. Specifically, the frequency-based method will recommend users' most frequently visited POIs. Miyamoto et al. [10] weighed the accuracy and diversity of recommendations according to the frequency of items. Besides, Collaborative Filtering (CF) is also a classical method in traditional POI recommendations. At present, many researchers have used the CF-based method to recommend POIs. Wasid et al. [11] proposed a new frequency counting method to capture context information. Then they applied context information to CF to improve similarity measurement. Wang et al. [12] used CF to find similar users. They not only considered the influence of geography and time but also considered the trust between users. This method improved the accuracy and recall of POI recommendations.

However, the methods based on CF are seriously affected by the cold start. It is difficult to get good recommendation results by using sparse user check-in data. Various researchers have made efforts to solve cold start. Liu et al. [13] tried to use the scores of other fields to alleviate the cold start. And Liu et al. [14] utilized Local Collaborative Ranking (LCR) for POI recommendation. Yin et al. [15] first expressed the user's personal preference hierarchically using geographical area information, which can improve the recommendation performance in the cold start scenario. However, these methods could alleviate the cold start problem, but they could not pay attention to the impact of time on user preferences. The traditional POI recommendation only predicts the POIs that users may like in the future but ignores the specific time information. For example, users may visit the recommended POIs immediately or five months later. This result will make the recommendation lack timeliness. No matter how to improve the accuracy of recommendations, the problem of timeliness cannot be solved.

B. Successive POI recommendation

With the wide application of sensors in the era of Internet of things [16] [17], successive POI recommendation has become a research hotspot in travelling enterprises. It needs to use the sequence information of users' check-ins fully. Therefore, some studies used HMM to capture the sequence information. However, due to the unfollow-up effect of HMM, the HMM-based methods could only capture users' short-term preferences but could not capture long-term preferences. Therefore, it is not effective in solving the problem of long-term sequence prediction. Deep learning methods have attracted extensive attention in the industry [18]. The method based on deep learning, mainly based on RNN, is trendy in sequence prediction [19]. Similarly, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are widely used as excellent variants of RNN. Although these methods can capture the long and short-term preferences of users, they can not capture the indirect relationship between users and POIs. Liu et al. [20] improved the above method to alleviate the information loss using bidirectional GRU to replace the basic GRU and added a time sliding window. Although these methods could improve the recommendation performance, they only considered limited influencing factors and did not consider group influence and privacy protection. Moreover, they only researched the POI category recommendation and did not further focus on the successive POI recommendation.

Recently, the method based on GNN has been widely used in the fields of recommendation systems and so on [21] [22]. Due to GNN can mine the high-order connectivity, it can genuinely improve the recommendation accuracy. Zhang et al. [23] used GNN to learn the representation of nodes from node information and topology and applied attention mechanism to learn heterogeneous social relations, which improved the

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3200067

accuracy of recommendation. Chen et al. [24] considered the heterogeneity and sparsity of check-in records, combined with the irregularity of time and distance to capture and learn complex sequential transformations. Wang et al. [25] used GNN to understand the different effects of different users and used hyperbolic translation to measure the rationality of recommendation. Although travelling enterprises have made a lot of efforts for successive POI recommendation [26], those methods still leave a lot to be desired. Precisely, the HMM-based method can not capture the long-term preferences. The RNN-based approach can not capture high-order connectivity. The GNN-based method ignores the dynamic timeliness of POIs. To overcome the above shortcomings, we have proposed ITGCN for POI recommendation. It can not only capture the dynamic timeliness of users and POIs but also capture the indirect relationship between users and POIs.

III. Approach

ITGCN can learn the dynamic embedding between users and POIs and capture the high-order connectivity. ITGCN mainly contains three components: the embedding, convolution, and prediction components, as illustrated in Figure 1. Details are as follows.

A. Definition

Definition 1. **Check-in:** U represents the user set in LBSN, and P represents the POI set. Then, we define the user check-in as $c_{u,p,t} = (u, p, b, t)$, representing the user u check-in at the POI p at time t. And the b is the interaction index in the ordered set of interactions.

Definition 2. **User-POI interaction graph:** A user-POI interaction graph is a bipartite graph G. Specifically, the vertex set of the graph contains users and POIs, and the edge set of the graph represents the interaction relationship between users and points-of-interest.

Definition 3. **User subset:** The user subset represents the latest interaction set of the user u. It includes the latest m interactions which can be shown as $S_{u,t} = \{c_{u,p,t_x} | p \in P, n - m < x \leqslant n\}$.

Definition 4. **POI subset:** The POI subset represents the latest interaction set of the POI p. It includes the latest m interactions which can be shown as $S_{p,t} = \{c_{u,p,t_x} | u \in U, n - m < x \leqslant n\}$.

Definition 5. **Successive POI recommendation in travelling enterprises:** Given the historical check-in records of target users, for example, the user u checked in at time t_M. We will predict the Top-K POIs that the user u may visit at t_{M+1}.

B. The Embedding Component

This component mainly generates four types of embedding: user embedding, point-of-interest embedding, time embedding, and interaction embedding. First, this component needs to generate user embedding. For each user, we train a d-dimensional vector $\mathbf{u}_{i,t}$ of each user u_i. So we can obtain the user embedding matrix $\mathbf{U_M} \in \mathbb{R}^{|U| \times d}$. Similarly, for each POI, we also train it as a d-dimensional vector $\mathbf{p}_{j,t}$, and then get a point-of-interest embedding matrix $\mathbf{P_M} \in \mathbb{R}^{|P| \times d}$, where $|U|$ indicates the total number of users, $|P|$ indicates the total number of POIs.

Next, this component needs to generate time embedding. Time is a critical factor in successive POI recommendations. However, time is difficult to express directly because of its dynamics. Inspired by [27] and [9], we adopt a specific time encoding technique. We define a continuous function $\phi(\cdot)$ to map the time intervals to d-dimensional vectors $\mathbf{t} \in \mathbb{R}^d$. In addition, we use the user subset and the POI subset to update user embedding $\mathbf{u}_{i,t}$ and time embedding $\mathbf{p}_{j,t}$.

Finally, we generate interaction embedding inspired by [28]. In the User-POI interaction graph, we first select the latest m interactions as the user subset (or the POI subset). Then, we use the information in the user subset (or the POI subset) to get interactive embedding $\mathbf{l}_{i,t} \in \mathbb{R}^d$ of $\mathbf{u}_i$ (or $\mathbf{l}_{j,t} \in \mathbb{R}^d$ of $\mathbf{p}_j$).

C. The Convolution Component

The convolution component's goal is to model the indirect relationship. For each embedding representation in the previous component, we first connect the embedding representations and then assign different weights to them. Then, we connect the output of the attention mechanism to a feed-forward neural network to introduce nonlinearity into the convolution layer. Finally, an attention network is connected to assign different weights to the output. After that, we stack multi-layer convolution to capture more neighbor information of users and POIs. Here, we update the embedding representation using the of users' subset and POIs' subset, which combines time and interaction influence. The specific details are as follows.

We use a GCN to learn the node representation in the User-POI interaction graph. For each node, we use the neighbor information (i.e., user subset or POI subset) to update the information of this node. Therefore, the node information will incorporate the hidden characteristics of neighbor information. Specifically, for the target user u at the b-th iteration, GCN will use his subset S_u to update the node information $\mathbf{o}_u^{(b)}$, and aggregate the subset feature vectors $o_{S_u}^{(b)}$. Details are as follows:

$$\mathbf{o}_{S_u}^{(b)} = \mathrm{AGGREGATE}\left(\left\{\mathbf{o}_{u'}^{(b-1)} \mid u' \in S_u\right\}\right) \tag{1}$$

$$\mathbf{o}_u^{(b)} = \sigma\left(\mathbf{w}^{(b)} \cdot \mathrm{CONCAT}\left(\mathbf{o}_u^{(b-1)}, \mathbf{o}_{S_u}^{(b)}\right)\right) \tag{2}$$

where $\mathbf{o}_u^{(b-1)}$ is the current representation of the node, $\mathbf{o}_u^{(b)}$ is the next update representation of the node, $\mathbf{o}_{S_u}^{(b)}$ is the subset representation of the node, $\mathrm{AGGREGATE}(\cdot)$ is an aggregation function, $\mathbf{w}^{(b)}$ is the weight matrix, and σ is sigmoid activation function.

However, GCN can capture the high-order connectivity by updating node information, but there are still two problems that traditional GCN cannot achieve. First, GCN cannot capture the interactive location information between users and POIs. In addition, GCN ignores the time influence, and then it does not have the ability to capture the sequence information. Therefore, we propose ITGCN model to capture the sequence information and generate the dynamic representation of nodes

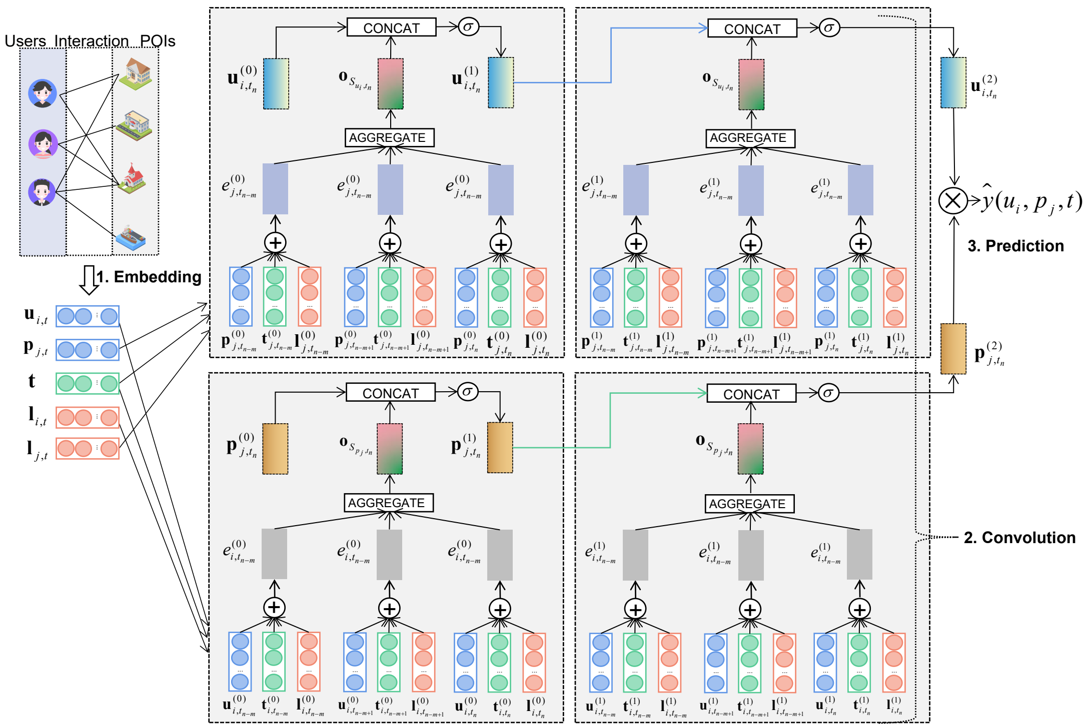


Fig. 1. Illustration of ITGCN for the successive POI recommendation task in travelling enterprises.

in G. Concretely, for the target user u_i and its interactive subset $S_{u,t} = \{c_{u,p,t_x} | p \in P, n-m < x \leqslant n\}$, GCN will use the subset $S_{u,t}$ to update the node information $\mathbf{o}^{(b)}_{s_{u_i,t_n}}$ by concatenating its current representation and aggregate the subset feature vectors $\mathbf{o}^{(b)}_{s_{u_i}}$. Details are as follows:

$$\mathbf{o}^{(b)}_{S_{u_i,t_n}} = \text{AGGREGATE}\left(\left\{\left(\mathbf{p}^{(b-1)}_{j,t}, \mathbf{t}, \mathbf{l}_{j,t}\right) \mid c_{u_i,p_j,t} \in S_{u_i,t_n}\right\}\right) \tag{3}$$

$$\mathbf{u}^{(b)}_{i,t_n} = \mathbf{w}_{U_2} \cdot \sigma\left(\mathbf{w}_{U_1} \cdot \text{CONCAT}\left(\mathbf{u}^{(b-1)}_{i,t_n}, \mathbf{o}^{(b)}_{s_{u_i,t_n}}\right)\right) \tag{4}$$

Here, $(\mathbf{p}^{(b-1)}_{j,t}, \mathbf{t}, \mathbf{l}_{j,t})$ is the representations of the POI nodes in $\mathbf{o}_{s_{u_i,t_n}}$ at the b-th iteration, $c_{u_i,p_j,t} \in S_{u_i,t_n}$ is the check-in of the target user u_i, $\mathbf{u}^{(b-1)}_{i,t_n}$ is the current representation of the u_i node, $\mathbf{u}^{(b)}_{i,t_n}$ is the subsequent update representation of the u_i node, $\mathbf{o}^{(b)}_{s_{u_i,t_n}}$ is the subset representation of the u_i node, $\mathbf{w}_{U_1}, \mathbf{w}_{U_2} \in \mathbb{R}^{d\times d}$ are the corresponding weight matrices. According to similar calculations, we can get the dynamic representation of each POI.

$$\mathbf{o}^{(b)}_{S_{p_j,t_n}} = \text{AGGREGATE}\left(\left\{\left(\mathbf{u}^{(b-1)}_{i,t}, \mathbf{t}, \mathbf{l}_{i,t}\right) \mid c_{u_i,p_j,t} \in S_{p_j,t_n}\right\}\right) \tag{5}$$

$$\mathbf{p}^{(b)}_{j,t_n} = \sigma\left(\mathbf{w}_{P_2} \cdot \text{CONCAT}\left(\mathbf{w}_{P_1} \cdot \mathbf{p}^{(b-1)}_{j,t_n}, \mathbf{o}^{(b)}_{S_{u_i,t_n}}\right)\right) \tag{6}$$

Here, $(\mathbf{u}^{(b-1)}_{i,t}, \mathbf{t}, \mathbf{l}_{i,t})$ is the representations of the user nodes in S_{p_j,t_n} at the b-th iteration, $c_{u_i,p_j,t} \in S_{p_j,t_n}$ is the check-in of the target POI p_j, $\mathbf{p}^{(b-1)}_{j,t_n}$ is the current representation of the p_j node, $\mathbf{p}^{(b)}_{j,t_n}$ is the next update representation of the p_j node, $\mathbf{o}^{(b)}_{S_{p_j,t_n}}$ is the subset representation of the p_j node, $\mathbf{w}_{P_1}, \mathbf{w}_{P_2} \in \mathbb{R}^{2d\times d}$ are the corresponding weight matrices.

To model the sequence information of user check-in, inspired by [2], we used the self-attention mechanism aggregator of aggregation function in the above calculation. Each aggregator has K self-attention layers, feed-forward layers, and vanilla attention layers. Details of the aggregator are as follows.

Step 1: For each check-in $c_{u_i,p_j,t} \in S_{u_i,t_n}$ at the b-th iteration, we first need to can get a whole representation. Specifically, we make a simple splicing of its POI embedding $\boldsymbol{p}^{(b)}_{j,t}$, time embedding $\boldsymbol{t}$ and interactive embedding $\mathbf{l}_{j,t}$. By this way, we can get a whole representation $e^{(b,0)}_j$.

$$e^{(b,0)}_j = \boldsymbol{p}^{(b)}_{j,t} + \boldsymbol{t} + \mathbf{l}_{j,t} \tag{7}$$

Then, through weight calculation, we can get the $e^{(b,k)}_j$ for each $\mathbf{p}_j$ (or $e^{(b,k)}_i$ for each $\mathbf{u}_i$) in the k-th layers. Here, the $a^{(b,k)}_{jr}$ can be calculated by using the softmax function.

$$e^{(b,k)}_j = \sum_{r=1}^{m} a^{(b,k)}_{jr} e^{(b,k-1)}_r \tag{8}$$

$$a^{(b,k)}_{jr} = \frac{\exp\left(Sim^{(b,k)}_{jr}\right)}{\sum_{r=1}^{m} \exp\left(Sim^{(b,k)}_{jr}\right)} \tag{9}$$

where the $Sim^{(b,k)}_{jr}$ is the similarity between POI_j and POI_r, which can be calculated by (10).

$$Sim^{(b,k)}_{jr} = \frac{e^{(b,k-1)}_j \cdot \left(e^{(b,k-1)}_r\right)^T}{\sqrt{d}} \tag{10}$$

Step 2: We use linear transformation to introduce non-linearity into the convolution layer. Here, W^k_1, W^k_2 represent

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3200067

the training parameter matrices.

$$f_j^{(b,k)} = FFN\left(e_j^{(b,k)}\right) \tag{11}$$

$$FFN\left(e_j^{(b,k)}\right) = \text{Layer Norm}\left(\text{Dropout}\left(\text{ReLU}\left(W_1^{(k)} \cdot e_j^{(b,k)}\right) W_2^{(k)} + e_j^{(b,k)}\right.\right. \tag{12}$$

Step 3: After getting the output $f_j^{(b,k)}$ of feed-forward neural network, we connect another attention layer. We still use the softmax function to calculate a weight coefficient $a_{ij}^{(b)}$.

$$a_{ij}^{(b)} = \frac{\exp\left(Sim_{ij}^{(b)}\right)}{\sum_{j=1}^{m}\exp\left(Sim_{ij}^{(b)}\right)} \tag{13}$$

where the similarity $Sim_{ij}^{(b)}$ between $U_{i,t}^{(b)}$ and $f_j^{(b,k)}$ is computed as follows:

$$Sim_{ij}^{(b)} = \frac{U_{i,t}^{(b)} \cdot \left(f_j^{(b,k)}\right)^T}{\sqrt{d}} \tag{14}$$

Finally, we can get the output $C_{S_{u_i},t}^{(b)}$ of the aggregator of S_{u_i,t_n}.

$$C_{S_{u_i},t}^{(b)} = \sum_{j=1}^{m} a_{ij}^{(b)} f_j^{(b,k)} \tag{15}$$

We can get the dynamic representation of users and POIs through the above operations, and capture their interaction information. In addition, we can stack multi-layer convolution to find more neighbor information between users and POIs. The algorithm for generating dynamic representation is shown in Algorithm 1.

D. The Prediction Component

According to the latest embedding representation incorporating dynamic time information and interactive information, we can predict the POI that users may visit next. In addition. We also design an optimization method to optimize the ranking of the recommendation list by using negative sampling. After getting the dynamic representation of users and POIs, we can make successive POI recommendation. We perform inner product operation between user embedding $z_{u_i,t}$ and POI embedding $z_{p_j,t}$, and finally get the recommendation result $\hat{y}(u_i, p_j, t)$.

$$\hat{y}(u_i, p_j, t) = (z_{u_i,t})^T \cdot z_{p_j,t} \tag{16}$$

Here, we have adopted the appropriate optimization method. Binary cross-entropy loss function can be well used. When the predicted value is close to the true value, the loss can be very small. When the predicted value is far away from the true value, the loss can be very large. This feature is conducive to the learning of the model. In order to optimize the recommendation order of interest points, we add negative POI p'_j and use negative sampling to optimize the order. It is defined as follows:

$$-\sum\nolimits_{<IT_{u_i,t},IT_{p_j,t},IT_{p'_j,t}>} \left[\log\left(\sigma\left(\hat{y}\left(u_i,p_j,t\right)\right)\right) + \log(1 - \sigma\left(\hat{y}\left(u_i,p'_j,t\right)\right)\right] + \lambda\|W\| \tag{17}$$

Algorithm 1 Embedding generation

Input: User-POI interaction graph $G =< (U,P)C >$, user set U, POI set P, check-in $c_{u,p,t} = (u,p,b,t)$, user subset $S_{u,t} = \{c_{u,p,t_x} | p \in P, n-m < x \leqslant n\}$, POI subset $S_{p,t} = \{c_{u,p,t_x} | u \in U, n-m < x \leqslant n\}$, depth B

Output: Dynamic representation of users $z_{u_i,t}$ and POIs $z_{p_j,t}$

if node in U **then**
  $US^{(B)} \leftarrow U_u S_{u,t}$
  for b = B to 2 **do**
    $US^{(b-1)} \leftarrow US^{(b)}$
    for each $c_{u_i,p_j,t}$ in $US^{(b)}$ **do**
      $US^{(b-1)} \leftarrow US^{(b-1)} \cup S(c_{u_i,p_j,t})$
    end for
  end for
  for b = 1 to B **do**
    for each $c_{u_i,p_j,t}$ in $US^{(b)}$ **do**
      Aggregate the POI nodes in $S_{u,t}$
      Update the embedding $\mathbf{u}_{i,t_n}^{(b)}$
      Get the output $z_{u_i,t}$
    end for
  end for
end if
if node in P **then**
  $PS^{(B)} \leftarrow P_p S_{p,t}$
  for b = B to 2 **do**
    $PS^{(b-1)} \leftarrow PS^{(b)}$
    for each $c_{u_i,p_j,t}$ in $PS^{(b)}$ **do**
      $PS^{(b-1)} \leftarrow PS^{(b-1)} \cup S(c_{u_i,p_j,t})$
    end for
  end for
  for b = 1 to B **do**
    for each $c_{u_i,p_j,t}$ in $PS^{(b)}$ **do**
      Aggregate the POI nodes in $S_{p,t}$
      Update the embedding $\mathbf{p}_{j,t_n}^{(b)}$
      get the output $z_{p_j,t}$
    end for
  end for
end if
return $z_{u_i,t}$ and $z_{p_j,t}$

Furthermore, The algorithm for training the ITGCN model is shown in Algorithm 2.

IV. EXPERIMENTS

A. Datasets, Evaluation Metrics and Baselines

We conducted experiments on four available datasets (i.e., Foursquare[1], Gowalla[2], NYC, and TKY [29]). In experiments, we removed unusual POIs with less than five user check-ins. Specifically, the detailed statistic information is illustrated in Table I.

We selected two common indicators to measure recommendation performance: Recall@K (R@k) and NDCG@K

[1]https://sites.google.com/site/yangdingqi/home/foursquare-dataset
[2]http://snap.stanford.edu/data/loc-gowalla.html

Algorithm 2 ITGCN model

Input: Training set $< IT_{u_i,t}, IT_{p_j}, t >$
Output: ITGCN model's parameter set
Initialize parameter set Δ
for each $c_{u_i,p_j,t}$ **do**
  Calculate the user embedding $z_{u_i,t}$
  Calculate the POI embedding $z_{p_j,t}$
  Calculate the negative sampling embedding $z_{p'_j,t}$
end for
Find Δ to minimize the Eq.17
return Δ

TABLE I
BASIC DATASET STATISTICS

	Users	POIs	Check-ins
Foursquare	7642	9989	179468
Gowalla	5628	26339	606220
New York City	1083	9989	179468
Tokyo	2293	15177	494807

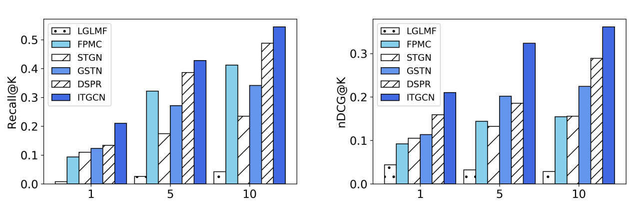
(N@K). Here, Recall@K can be used to measure the proportion of POIs accurately recommended, and NDCG@K can be used to measure the ranking performance.

$$\mathrm{R}\,@k = \frac{1}{|U|}\sum_{u=1}^{|U|}\frac{|\mathrm{F}_u(k)\cap \mathrm{True}_u|}{|\mathrm{True}_u|} \tag{18}$$

$$N@k = \frac{1}{|U|}\sum_{u=1}^{|U|}\frac{1}{Z_u}\sum_{j=1}^{k}\frac{2^{I(|\{f_u^j\}\cap True_u|)}-1}{\log_2(j+1)} \tag{19}$$

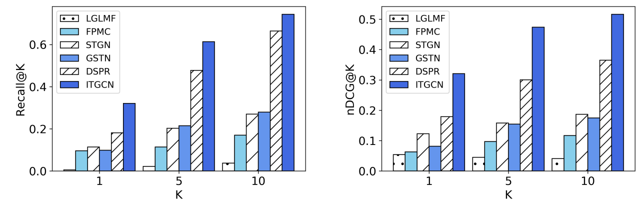
where $F_u(k)$ donates the recommended POIs, $True_u$ donates the POIs of the user who will check-in, where f_u^j donates the j-th recommended POI in $F_u(k)$, $I(\cdot)$ denotes the indicator function and Z is a normalized constant that is the maximum value of DCG@k.

We compare ITGCN with other advanced models. Specifically, we compared ITGCN with LGLMF [30], which used matrix decomposition to mine the spatial information of POIs, so as to make recommendations. Next, we compare ITGCN with FPMC [31], which used Factorized Personalized Markov Chain to mine users' location preferences. We also compared ITGCN with STGN [32], GSTN [33] and DSPR [34], which used improved LSTM, Graph-enhanced Spatial-Temporal network, attention-based LSTM respectively to make POI recommendations. Finally, we visualized the comparison results in Fig 2 and Fig 3.



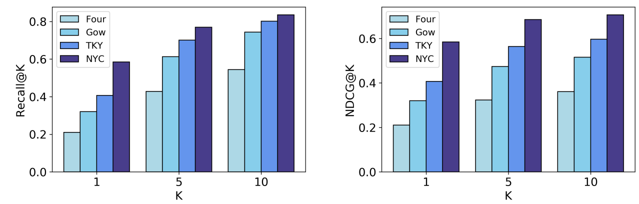
(a) (b)

Fig. 2. Comparison of recommendation results in Foursquare.



(a) (b)

Fig. 3. Comparison of recommendation results in Gowalla.



(a) (b)

Fig. 4. Recommendation results of ITGCN in four datasets.

B. Experimental Results and Discussions

In the experiment, we choose k = 1, 5 and 10 as the recommended number. k = 1 is selected to provide users with unique choices, and k = 5 and k = 10 are selected to provide users with diverse choices. The comparison results between ITGCN and other models in Foursquare and Gowalla are listed in Table II and visualized in Fig 2 and Fig 3. It shows that when the number of recommendations is 1, 5 and 10, the recall and nDCG of ITGCN are the best. For example, compared with the DSPR [34] model, the recall of ITGCN increased by 7.63%, 4.18% and 5.68% respectively, and the nDCG of ITGCN increased by 5.11%, 13.8% and 7.23% respectively in the Foursquare dataset. Similarly, the recall of ITGCN increased by 13.92%, 13.6% and 7.93% respectively, and the nDCG of ITGCN increased by 14.14%, 17.37% and 15.08% respectively in the Gowalla dataset. Compared with the GSTN model, the recall of ITGCN increased by 8.68%, 15.65% and 20.27% respectively, and the nDCG of ITGCN increased by 9.64%, 12.16% and 13.66% respectively in the Foursquare dataset. Similarly, the recall of ITGCN increased by 22.19%, 39.85% and 46.41% respectively, and the nDCG of ITGCN increased by 23.86%, 31.97% and 34.1% respectively in the Gowalla dataset. This is because ITGCN can not only capture the high-order connectivity between users and POIs, but also capture the time impact between users and POIs.

We show the comparison of recall and nDCG in the Foursquare dataset in Fig 2. It can be seen that the recall and nDCG of ITGCN model is the highest. Similarly, we show the comparison of recall and nDCG in the Gowalla dataset in Fig 3. In addition, we also visualize the recommended performance of ITGCN in four datasets in Fig 4 to reflect the impact of different datasets on ITGCN. In Fig 4, ITGCN has the best performance in foursquare, followed by Gowalla, TKY and NYC. This shows that ITGCN is indeed affected by the dataset, and the more data, the better the recommendation performance of ITGCN. However, our model can still main-

TABLE II
THE COMPARISON OF EXPERIMENTAL RESULTS

	Foursquare						Gowalla					
	R@1	N@1	R@5	N@5	R@10	N@10	R@1	N@1	R@5	N@5	R@10	N@10
LGLMF	0.0077	0.0443	0.0263	0.0325	0.0427	0.0286	0.0054	0.0542	0.0221	0.0449	0.0376	0.0406
FPMC	0.0937	0.0921	0.3219	0.1439	0.4120	0.1547	0.0955	0.0631	0.1139	0.0967	0.1698	0.1167
STGN	0.1098	0.1052	0.1743	0.1324	0.2345	0.1558	0.1139	0.1227	0.2029	0.1579	0.2698	0.1866
GSTN	0.1233	0.1137	0.2713	0.2019	0.3416	0.2246	0.0984	0.0817	0.2146	0.1541	0.2791	0.1749
DSPR	0.1338	0.1590	0.3860	0.1855	0.4875	0.2889	0.1811	0.1789	0.4771	0.3001	0.6639	0.3651
ITGCN	**0.2101**	**0.2101**	**0.4278**	**0.3235**	**0.5443**	**0.3612**	**0.3203**	**0.3203**	**0.6131**	**0.4738**	**0.7432**	**0.5159**

tain relatively high recommendation performance even in the smallest dataset (i.e., NYC dataset), which also proves that ITGCN has the advantages of good universality.

The method we proposed that considered some factors that other researchers have not considered, but there are also some influencing factors have not been included in our research. We select some factors to summarize, and the results are briefly shown in Table III. Our summary and analysis of advantages and disadvantages are as follows. For the advantages, the ITGCN we proposed can both consider the time influence and interaction influence. It can pay attention to the changes of user preferences over time, and can mine the indirect interactive information between users and POIs (i.e., High-order connectivity, Hoc). For the disadvantages, in the experiments, we delete the POIs with less than 5 check-in times to alleviate the negative impact of data sparsity. Our goal is not to solve the cold start problem. In addition, we don't pay attention to the privacy protection in successive point-of-interest recommendation. We default the model we proposed is used by a single smart enterprise, and there is no data interaction between smart enterprises. In the era of big data, information interaction between smart enterprises may be unavoidable. In the future, we will also try to make successive POI recommendation that takes the privacy protection and high precision into account.

TABLE III
INFLUENCING FACTORS CONSIDERED IN ITGCN MODEL

	Time	Hoc	Interaction	Cold-start	Privacy
ITGCN	√	√	√	×	×

C. Parameter Selection

- Embedding dimension. Setting appropriate embedding dimension can improve the performance of the model. When converting a graph node into a vector representation, we need to specify the dimension of the vector. Because the vector dimension will influence recommendation results. In this experiment, we list four candidates: 120, 160, 200, and 240. With the increase of embedding dimension, we try to find embedding dimension that can achieve better prediction performance of the model. Later, we train the performance of ITGCN on four public datasets. The training results are visualized in Fig 5. With the increase of embedding dimension, the recall and nDCG of ITGCN also increase, but when the dimension exceeds 200, the performance of ITGCN will decline. This phenomenon may be caused by over fitting. Finally, the embedding dimension we selected for ITGCN is equal to 200.

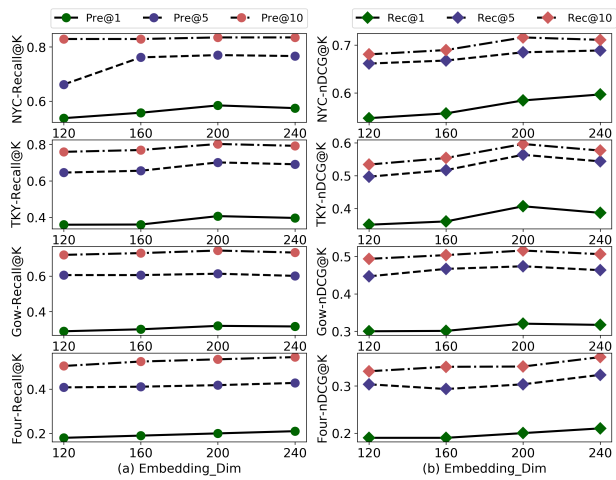


Fig. 5. Results of different embedding dimensions.

- Self-attention layer. Properly increasing the attention layer can make ITGCN have better expression ability, whereas too many layers of attention will also make the calculation too complex. In this experiment, we give four candidates: 0, 1, 2, and 4. Then the results of ITGCN on the four public datasets are visualized in Fig 6. When the number of layers is 1, the performance of the ITGCN is greatly improved, and then the performance of the model tends to be stable. Therefore, the number of self-attention layers we selected for ITGCN equals 1.

V. CONCLUSION

This paper proposed a novel model (i.e., ITGCN) for Successive Point-of-Interest Recommendation in travelling enterprises. Experiments on four datasets prove the effectiveness of ITGCN in augmented Intelligence of Things for smart enterprise systems. In our future work, we hope to make successive POI recommendation that takes the privacy protection and high precision into account. We may try to use Locality-Sensitive Hashing, learning to hash and other methods to carry out our next work.

VI. ACKNOWLEDGMENT

his research is supported by the National Key Research and Development Program of China (No.2020YFB1707600)

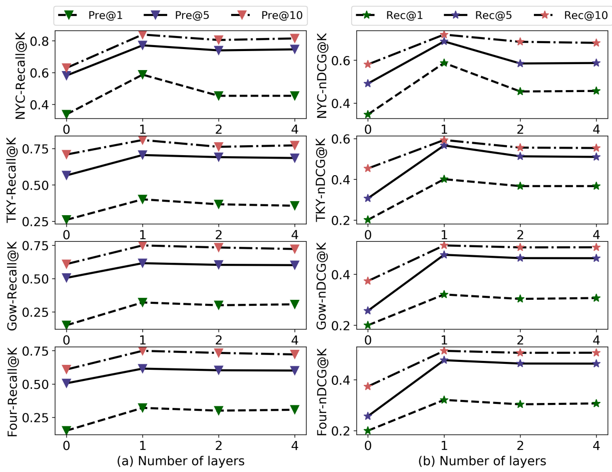


Fig. 6. Results of different self-attention layers.

and the National Natural Science Foundation of China (No. 61872219 and No. 62177014).

References

[1] S. Hu, Y. Shi, A. Colombo, S. Karnouskos, and X. Li, "Guest editorial: cloud-edge computing for cyber-physical systems and internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7802–7805, 2021.

[2] A. Rana, C. Chakraborty, S. Sharma, S. Dhawan, S. K. Pani, and I. Ashraf, "Internet of medical things-based secure and energy-efficient framework for health care," *Big Data*, vol. 10, no. 1, pp. 18–33, 2022.

[3] L. Qi, C. Hu, X. Zhang, M. R. Khosravi, S. Sharma, S. Pang, and T. Wang, "Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4159–4167, 2021.

[4] Y. Sun, J. Liu, K. Yu, M. Alazab, and K. Lin, "Pmrss: privacy-preserving medical record searching scheme for intelligent diagnosis in iot healthcare," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1981–1990, 2021.

[5] H. Liu, C. Zheng, D. Li, X. Shen, K. Lin, J. Wang, Z. Zhang, Z. Zhang, and N. N. Xiong, "Edmf: Efficient deep matrix factorization with review feature learning for industrial recommender system," *IEEE Transactions on Industrial Informatics*, 2021, doi: 10.1109/TII.2021.3128240.

[6] K. Chadaga, C. Chakraborty, S. Prabhu, S. Umakanth, V. Bhat, and N. Sampathila, "Clinical and laboratory approach to diagnose covid-19 using machine learning," *Interdisciplinary Sciences: Computational Life Sciences*, pp. 1–19, 2022.

[7] J. B. Awotunde, S. Oluwabukonla, C. Chakraborty, A. K. Bhoi, and G. J. Ajamu, "Application of artificial intelligence and big data for fighting covid-19 pandemic," *Decision Sciences for COVID-19*, pp. 3–26, 2022.

[8] S. Kumar, X. Zhang, and J. Leskovec, "Predicting dynamic embedding trajectory in temporal interaction networks," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 1269–1278.

[9] L. Huang, Y. Ma, Y. Liu, S. Wang, and D. Li, "Position-enhanced and time-aware graph convolutional network for sequential recommendations," *arXiv preprint arXiv:2107.05235*, 2021.

[10] S. Miyamoto, T. Zamami, and H. Yamana, "Appearance frequency-based ranking method for improving recommendation diversity," in *2019 IEEE 4th International Conference on Big Data Analytics (ICBDA)*. IEEE, 2019, pp. 420–425.

[11] M. Wasid, V. Kant, and R. Ali, "Frequency-based similarity measure for context-aware recommender systems," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2016, pp. 627–632.

[12] W. Wang, J. Chen, J. Wang, J. Chen, J. Liu, and Z. Gong, "Trust-enhanced collaborative filtering for personalized point of interests recommendation," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6124–6132, 2019.

[13] W. Liu, J. Su, C. Chen, and X. Zheng, "Leveraging distribution alignment via stein path for cross-domain cold-start recommendation," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[14] W. Liu, H. Lai, J. Wang, G. Ke, W. Yang, and J. Yin, "Mix geographical information into local collaborative ranking for poi recommendation," *World Wide Web*, vol. 23, no. 1, pp. 131–152, 2020.

[15] H. Yin, W. Wang, H. Wang, L. Chen, and X. Zhou, "Spatial-aware hierarchical collaborative deep learning for poi recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 11, pp. 2537–2551, 2017.

[16] P. Zhang, L. Li, K. Niu, Y. Li, G. Lu, and Z. Wang, "An intelligent wireless transmission toward 6g," *Intelligent and Converged Networks*, vol. 2, no. 3, pp. 244–257, 2021.

[17] C. Catlett, P. Beckman, N. Ferrier, H. Nusbaum, M. E. Papka, M. G. Berman, and R. Sankaran, "Measuring cities with software-defined sensors," *Journal of Social Computing*, vol. 1, no. 1, pp. 14–27, 2020.

[18] R. Gu, Y. Chen, S. Liu, H. Dai, G. Chen, K. Zhang, Y. Che, and Y. Huang, "Liquid: Intelligent resource estimation and network-efficient scheduling for deep learning jobs on distributed gpu clusters," *IEEE Transactions on Parallel and Distributed Systems*, 2021, doi: 10.1109/TPDS.2021.3138825.

[19] Y. Liu, D. Li, S. Wan, F. Wang, W. Dou, X. Xu, S. Li, R. Ma, and L. Qi, "A long short-term memory-based model for greenhouse climate prediction," *International Journal of Intelligent Systems*, vol. 37, no. 1, pp. 135–151, 2022.

[20] Y. Liu, Z. Song, X. Xu, W. Rafique, X. Zhang, J. Shen, M. R. Khosravi, and L. Qi, "Bidirectional gru networks-based next poi category prediction for healthcare," *International Journal of Intelligent Systems*, 2021, doi: 10.1002/int.22710.

[21] X. Song, J. Li, Q. Lei, W. Zhao, Y. Chen, and A. Mian, "Bi-clkt: Bi-graph contrastive learning based knowledge tracing," *arXiv preprint arXiv:2201.09020*, 2022.

[22] B. M. Oloulade, J. Gao, J. Chen, T. Lyu, and R. Al-Sabri, "Graph neural architecture search: A survey," *Tsinghua Science and Technology*, vol. 27, no. 4, pp. 692–708, 2021.

[23] J. Zhang, X. Liu, X. Zhou, and X. Chu, "Leveraging graph neural networks for point-of-interest recommendations," *Neurocomputing*, vol. 462, pp. 1–13, 2021.

[24] Y. Chen, T. Thaipisutikul, and T. K. Shih, "A learning-based poi recommendation with spatiotemporal context awareness," *IEEE Transactions on Cybernetics*, 2020, doi: 10.1109/TCYB.2020.3000733.

[25] H. Wang, D. Lian, H. Tong, Q. Liu, Z. Huang, and E. Chen, "Hypersorec: Exploiting hyperbolic user and item representations with multiple aspects for social-aware recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 40, no. 2, pp. 1–28, 2021.

[26] P. Nitu, J. Coelho, and P. Madiraju, "Improvising personalized travel recommendation system with recency effects," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 139–154, 2021.

[27] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," *arXiv preprint arXiv:2002.07962*, 2020.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[29] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 129–142, 2014.

[30] H. A. Rahmani, M. Aliannejadi, S. Ahmadian, M. Baratchi, M. Afsharchi, and F. Crestani, "Lglmf: local geographical based logistic matrix factorization model for poi recommendation," in *Asia Information Retrieval Symposium*, 2019, pp. 66–78.

[31] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 811–820.

[32] P. Zhao, A. Luo, Y. Liu, F. Zhuang, J. Xu, Z. Li, V. S. Sheng, and X. Zhou, "Where to go next: A spatio-temporal gated network for next poi recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2020, doi: 10.1109/TKDE.2020.3007194.

[33] Z. Wang, Y. Zhu, Q. Zhang, H. Liu, C. Wang, and T. Liu, "Graph-enhanced spatial-temporal network for next poi recommendation," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2022, doi: 10.1145/3513092.

[34] H. Wang, P. Li, Y. Liu, and J. Shao, "Towards real-time demand-aware sequential poi recommendation," *Information Sciences*, vol. 547, pp. 482–497, 2021.

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3200067

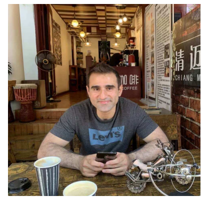
Yuwen Liu received the master's degree from the School of Computer Science, Qufu Normal University, China, in 2022. She is currently a Ph.D. Candidate with the College of Computer Science and Technology, China University of Petroleum, China. Her research interests include recommender systems and privacy protection.

Huiping Wu received the B.S. degree from the School of Computer Science and Technology, Shandong University of Technology, China, in 1994 and the M.E. degree from the School of Computer Science and Technology, Ocean University of China, China, in 2012. Since 1998, she has been a lecturer in Weifang University of Science and Technology, China. She has published many research papers in reputable journals and conferences. Her research interests include computer application technology, cloud computing and internet of things.

Khosro Rezaee graduated in 2014 and 2018 from Hakim Sabzevari University, Sabzevar, Iran, with master's and doctorate degrees in biomedical engineering, respectively. He is currently an assistant professor at Meybod University in Meybod, Iran. He focuses on machine learning, signal processing, deep learning, medical image processing, and computer assisted diagnosis in his research.

Mohammad R. Khosravi is an international research fellow with Shandong Provincial University Laboratory for Protected Horticulture, Weifang University of Science and Technology, China, and a research associate at Department of Computer Engineering, Persian Gulf University, Iran. He completed his academic research to earn a PhD degree in Communications Engineering (EECS) at Shiraz University of Technology, Iran, from 2016 to 2020. Later on, he joined the industry sector to develop some related industrial research from 2020 to 2022 as a part of his scientific commitments. Mohammad has also received his BSc and MSc degrees in Electrical Engineering from Shiraz University in 2013, and Persian Gulf University in 2015, both based at Iran, respectively. His main topics of interest are multimedia systems, high-performance computing and communications.

Osamah Ibrahim Khalaf received the B.Sc. degree in software engineering field from Al-Rafidain University College, Iraq, the M.Sc. degree in computer engineering field from Belarussian National Technical University, and the Ph.D. degree in the field of computer networks from the Faculty of Computer Systems and Software Engineering, University Malaysia Pahang. He is currently a Senior Engineering and Telecommunications Lecturer with Al-Nahrain University.



Arif Ali Khan works as an Assistant Professor with the M3S Unit, Empirical Software Engineering in Software, Systems and Services, University of Oulu, Finland. Previously he worked as a faculty member with the Faculty of Information Technology, University of Jyvaskyla, Finland; College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China; and Department of Computer Science, COMSATS University, Islamabad, Pakistan. He received a BS degree in software engineering from the University of Science and Technology, Bannu, Pakistan and an MSc degree in information technology from Universiti Teknologi PETRONAS, Malaysia. He obtained a PhD degree in software engineering from the Department of Computer Science, City University of Hong Kong. He has participated in and managed several empirical software engineering-related research projects.

Dharavath Ramesh is a Professor (Asst.) in the Computer Science and Engineering Department at Indian Institute of Technology Dhanbad, Jharkhand, India. He obtained Ph.D from Indian Institute of Technology (Indian School of Mines), Dhanbad in 2015. His Master's degree in Computer Science and Engineering with a specialization as Software Engineering is from Jawaharlal Nehru Technological University, Hyderabad, India in 2009. His Bachelor of Engineering degree in Computer Science and Engineering is from KITS, Warangal, India in 2004. He works and publishes widely in the areas of Blockchain and Distributed Computing, Distributed Databases, Cloud Databases, Modelling Big Data, Processing Big Data, Virtualization and Scheduling in Cloud Environment, Brain Computer Interaction, Community Detection in Social Networks, ML paradigms in Agriculture.

Lianyong Qi received his PhD degree in Department of Computer Science and Technology from Nanjing University, China. In 2010, he visited the Department of Information and Communication Technology, Swinburne University of Technology, Australia. Now, he is a professor of the College of Computer Science and Technology, China University of Petroleum, China. His research interests include big data, recommender systems and service computing.