Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

NPR-LBN: next point of interest recommendation using large bipartite networks with edge and cloud computing

Inayat Khan[1], Anwar Sadad[2], Gauhar Ali[3*], Mohammed ElAffendi[3], Razaullah Khan[1] and Tariq Sadad[1]

Abstract

During the last decades, tourism has been augmented worldwide through which the diversity of tourists' interests is increased and is challenging to tackle with the traditional management system. Such challenges can be overcome by LBSNs (Location-Based Social Networks) such as Yelp, Foursquare, and Facebook which help to collect more personalized information close to tourists' preferences/interests like check-ins, comments, and reviews. In this regard, solutions have been proposed to exploit the POI (Point of Interest) recommendation, but they failed to overcome sparsity and cold-start problems. Existing methods are also not focusing on important aspects, including geographical context, dynamics preferences and social influence, which are essential factors in POI recommendation. Therefore, this work tried to incorporate these factors and present a unified model using bipartite networks to learn users and POI dynamics. For this purpose, we have represented all the factors using eleven networks and combined them into a single latent space. In addition, Edge Computing processes data at the network's edge, reducing latency and bandwidth usage and enabling real-time and personalized recommendations. Furthermore, cloud computing could be used to store and process the large amounts of data collected from LBSNs, to support the proposed model's computational requirements and make it more accessible and scalable, allowing it to be easily used by tourism management systems worldwide. Experimental results show that our model outperforms state-of-the-art methods using real-world dataset in terms of accuracy and perform better against sparsity and cold-start problems.

Keywords Tourism, Recommender systems, Location graph, Check-in history dataset

Introduction

"The United Nations World Tourism Organization (UNWTO)" reported that since 1950 the number of tourist arrivals has been raised to 1.4 billion yearly. Also, the fast growth in tourism is expected to reach 1.8 billion worldwide by the year 2030. Tourism plays a vital role in extending economic freedom in developed countries and presents a paradox. To overcome this paradox, different companies related to the tourism sector can play a vital role in different sectors, such as business communities and industries. In the past decade, a significant improvement has been witnessed by development experts, industry leaders, and policymakers toward the tourism sectors in various countries in the world [1]. Consequently, tourism has gained positive economic outcomes, especially by boosting the GDP (Gross Domestic Product) and providing employment opportunities [2]. Considering the growth of tourism and travellers' necessities, it is pertinent to enhance the services provided to travelers according to their needs and interests [3]. Therefore, exploiting the choices and preferences of users is a hot

*Correspondence:
Gauhar Ali
gali@psu.edu.sa
[1] Department of Computer Science, University of Engineering and Technology, Mardan 23200, Pakistan
[2] Quaid-I-Azam University, Islamabad, Pakistan
[3] EIAS Data Science and Blockchain Lab, College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia

topic in academia as it greatly impacts decision-making, decision rules, and choice factors [4]. On the contrary, acute developments in the web, social networks, big data [5], cloud computing, IoVs (Internet of Vehicles) [6, 7], and IoTs (Internet of Things) technologies provide abysmal information that acquaints information overload problems. The individuals are precarious in choosing relevant information and making decisions. Therefore, recommender systems in information technology come in, which cope with the information overload problem [8]. It suggests relevant information to the users, considering their explicit or implicit preferences. Therefore, computer scientists contributed to the tourism industry, and plenty of research has been conducted to facilitate tourists using recommender systems. Due to dynamic and temporal preferences, the existing approaches are limited to coping with the sparsity [9] and cold-start [10] problems. Extensive research [11–18] have been devoted to this area which focuses on users and location relationship, but they exclude sparsity problem due to preference dynamics. In this regard, studies [19–22] have collected data about the relationship between users and tied it to user location but failed to resolve dynamic and temporal preferences. Besides, research dealing with users' temporal dynamics is still limited in alleviating the sparsity problem since these models do not exploit auxiliary contextual information which changes with user preferences over time [10]. For simplicity let's say a tourist loves to visit mountains in summer and cities in winter, providing them with mountains in winter and cities in summer will be inappropriate or irrelevant. Therefore, the proposed model temporal factor by splitting the dataset into seasons and categories of locations. Using this approach, user satisfaction with recommendations will be enhanced and can achieve higher accuracy respectively.

To prudently overcome aforesaid issues, there is a need for a unified model that exploits the behavior and preference dynamics of users for a more personalized recommendation. Therefore, the proposed model alleviates such problems and makes the following contributions.

- The proposed model uses an approach based on a probabilistic weighting strategy using eleven graphs to tackle the sparsity problem.
- Presents two algorithms to get users' favorite season(s) with the most visited categories in a particular season using past check-ins history.
- The proposed approach uses the work of RELINE (Recommendation with Multiple Network Embeddings) [10] and tries to learn the embeddings of graphs by using the concept of graphs to find the heterogeneous preferences of users.

The rest of the paper is organized as follows: Literature review section presents the contributions being devoted to POI recommendations. Participated networks in the recommendation model section emphasizes participated networks in the proposed model, Proposed next-POI recommender system section explains the proposed work, and Results and discussion section discusses obtained results. Finally, Conclusion and future work section concludes the proposed work and presents future work.

Literature review

This section discusses the contribution being devoted to recommender systems facilitating POIs. In [21] the authors have proposed a system using Hadoop technology which consists of four phases; scrapping, mapping, de-duplication, and recommendation. The shortcoming of this method is a user-centric approach. Due to its complexity, it increases computation time. Likewise, [23] proposed a collaborative filtering approach that performs better due to WSN (Wireless Sensor Network) [24, 25] installations around tourist sites (IoT sensors on edge). It provides the convenience of uploading tourist information and rating POI using smartphones. However, such a method fails to tackle the cold start problem because it is not feasible to implement WSN in all tourist spots, as a result, there may be some locations that remain unrated/unvisited due to the unavailability of sensors/devices at various spots. In this regard, [14] and [25] come up with different approaches to resolving the cold start problem using the notion of CARS (Context-Aware Recommender System), they have tried to get contextual information for achieving better results but ignored the importance of preference dynamics. Sampling on graphs has been used in various flavors, but less attention has been paid to matching a large set of graph properties. To this end, various studies employed network embedding models to exploit semantic relations between the network objects and generate their low-dimensional representations. In [26] the authors proposed a graph-based POI recommendation incorporating geographical and temporal influence to tackle the cold-start problem, but they ignored the importance of preference dynamics. Similarly, [27] have considered users' preference dynamics but ignored social influence. Furthermore, [28] realized the need to provide POI recommendations at an appropriate time rather than only exploiting user, social, and geographical preferences. Finally, authors in [15] upgraded the work of LINE (Large-scale Information Network Embedding) [29] and used large bipartite graphs to cope with cold start problems achieving good accuracy using social-, geographical-, temporal-influence, along with users' preference dynamics. To clarify, Table 1 summarizes the

Table 1 Incorporating factors in existing approaches

#	Approaches	Factors					
		Soc	Tem	Spa	PD	LC	Sea
1	K. A. Achmad *et al.* [9]						
2	G. Christoforidis *et al.* [10]	✓	✓	✓	✓		
3	M. Xie *et al.* [26]		✓	✓			
4	M. Xie *et al.* [27]		✓	✓	✓		
5	Y. Liu *et al.* [28]	✓	✓				
6	N. Joorablo *et al.* [30]	✓	✓		✓		
7	F. Ding *et al.* [31]	✓	✓	✓			
8	**NPR-LBN**	✓	✓	✓	✓	✓	✓

Abbreviations: Soc Social, *Tem* Temporal, *Spa* Spatial, *PD* Preference Dynamic, *LC* Locations Categorization, *Sea* Seasons

incorporating factors of existing methods while exploiting POI recommendations.

To exploit users' or tourists' behavioral patterns, we have highlighted the limitations of existing methods, many models [13, 28, 30–32] employed POIs as nodes and don't consider the spatial dimension with distance information. However, [13, 19, 28] have considered location influence but ignore the preference dynamics, which change over time. Methods [31–33] used to capture the temporal dynamics elegantly but do not incorporate spatial dimension. Furthermore, methods that tackle spatial and temporal behavior but fail to include preference evaluation, and finally, methods that capture all factors failed to maintain users' satisfaction regarding recommendations.

Participated networks in the recommendation model

This section presents the proposed model's workflow and discusses the participated networks and problem definition. Figure 1 depicts all incorporated networks in the proposed model, where Table. 2 describes mostly used symbols in the text.

As depicted in Fig. 1, eleven graphs (unipartite and bipartite) have been used in the proposed model. The proposed model consists of the social and location layers, which utilize all the incorporated graphs. The social layer represents the relationship between users (friendship) with each other. Similarly, the location layer describes the physical relationship between various locations like distance, height, and temperature. The embeddings have been generated for each graph and have fed to a collective space, where all the graphs are combined into a single vector space. To further understand the proposed model, the subsequent sub-sections explain the role of each participated graph as follows.

Point-of-Interest

Point-of-Interest is the location where tourists can take interest and have most check-in. It can be represented as a set: (s_{id}, lon, lat), where s_{id} specifies a location in the dataset, lon and lat refer to the longitude and latitude of a particular location.

Check-in

It is the presence of a user u in desired place l at a particular instance of time t, denoted as $\boldsymbol{c_i = (u, l, t)}$. A user u can check in only one place where they can record multiple check-ins in their profile $\boldsymbol{c_{ui} = \left\{(l_i, t_i), \ldots, (l_j, t_j)\right\}}$. For each user, a profile is maintained that stores the locations being visited by him/her; as the user profile grows, the preferences of a particular user will be more helpful in the recommendation.

Season

It can be defined as the season of the dataset and is divided into four seasons (Winter, Summer, Spring, and Autumn). Each season has the check-ins of all tourists during season ΔS. Every tourist prefers to visit or go for a tour in a particular season, while some tend to go for a visit in a few seasons or even all seasons, respectively. The importance of this season must be realized in the POI recommendation. Therefore, we have used time as a season.

User-user graph

The social interaction between users can be represented by and User-user graph as it is a unipartite graph. It can be represented by$G_{uu} = (U \cup V, W_{uv})$, where U and V

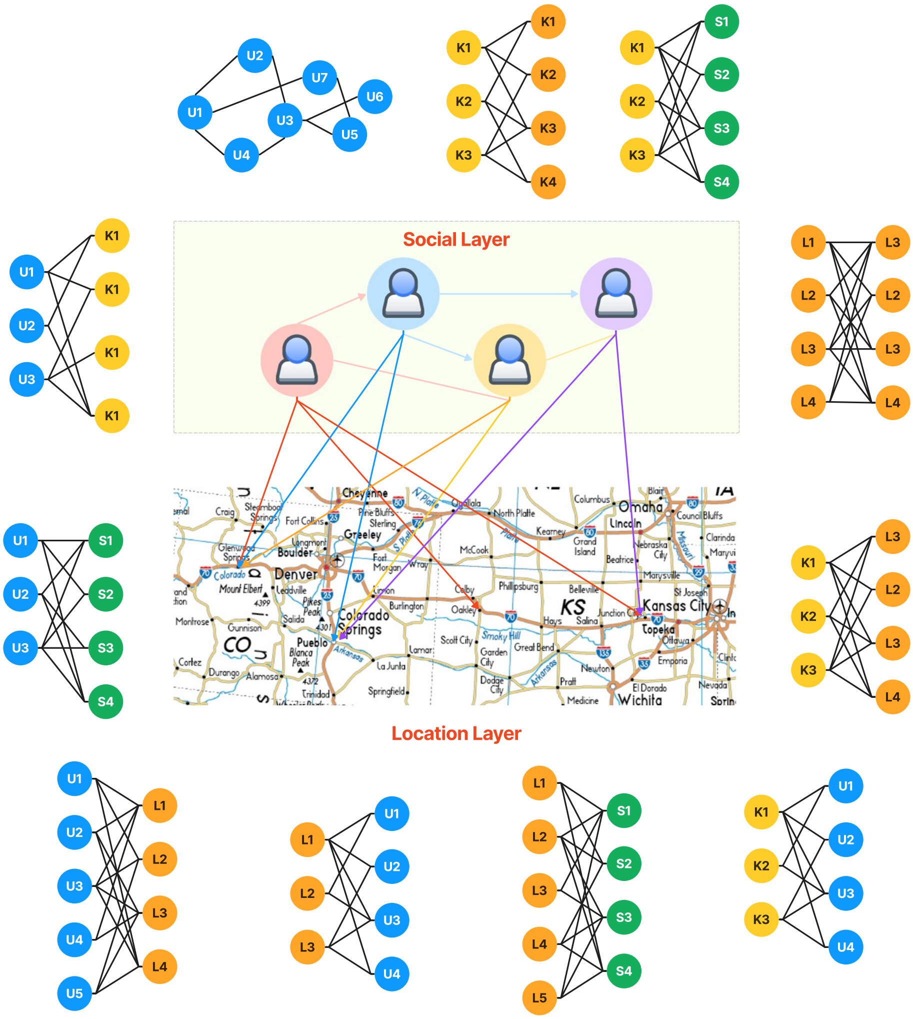

Fig. 1 All participating networks in the proposed model

Table 2 Commonly used symbols in the text

Symbol	Description
$c_{u,i}$	User's check-ins
ΔS	Season
L	Location
K	Location Category
W	List of weights $w_{i,j}$ on each graph

represent the sets of users $, while W_{uv}$ describes the set of weights among users in the network, which can be computed by Eq. (1) as follows

$$W_{uv} = \frac{1}{\sum_{i=1}^{n} |v_i|} \tag{1}$$

User-category graph

A bipartite graph shows the user's and category's connection, considering an entire check-in history. Specifically, it shows the significance of a specific category against all categories for a candidate user. Symbolically, this graph is represented by $G_{uk} = (U \cup K, W_{uk})$, in which $U\ and\ K$ are set of users and categories. W_{uk} is a set of weighted edges between U and K which can be computed using Eq. (2) and indicates the number of check-ins made in desired category k_i against overall check-ins made by user $u_{i.}$

$$W_{uk} = \frac{\sum_{\forall c_{u_i} \in k_i} c_{u_i,k}}{\sum_{\forall c_{u_i} \in K} c_{u_i,K}} \tag{2}$$

User-season graph

A bipartite graph represents the relationship between user u_i and season s_i. Algorithm 1 has been used to compute the significance of season s_i for each user u_i. Can be denoted as $G_{us} = (U \cup S, W_{us})$, in which S and U denote the set of seasons set users. W_{us} ia s set of weighted links as computed by Eq. (3), the number of user's check-ins in overall season check-ins made by the user u_i.

$$W_{us} = \frac{\sum_{\forall c_{u_i} \in s_i} c_{u_i,s}}{\sum_{\forall c_{u_i} \in S} c_{u_i,S}} \tag{3}$$

```
Inputs
C: Users' check-in history
U: List of users
S: List of seasons

Output
L: List of most visited seasons

1  Sorting all check-ins in C based on date
2  Create a list of the most favorite seasons of users $L$.

3  for $i \leftarrow 0$ to $lenghtof(S)$
4    if ($c_{ui}[i]$ in $S[i]$) then
5      $L_{ui}$ [$S$ [i]] += 1
6  return L
```

Algorithm 1. Extracting favorite season(s) of each user

Algorithm 1 tries to extract a list of seasons that the user visits most. Checkin history, users, and seasons (winter, summer, spring, autumn) are provided as inputs. In step 1 the check-ins have been sorted on timestamp/date, whereas in step 2 create a list L for the most visited season(s). Step 3 runs for each season, and step 4 checks whether the user u_i checked-in in S_i. If it is true, step 5 increment the season by one and assign it to user u_i in list L. Finally, step 6 returns the obtained L that consists of users and their favorite season(s).

User-location graph

A bipartite graph represents the degree of a specific location for a given user in a desired category. This relation is represented by $G_{ul} = (U \cup L, W_{ul})$ U and L denote the sets of users and locations respectively. This relation is represented as $G_{ul} = (U \cup L, W_{ul})$. Equation (4) computes the number of times one user visited a particular location in a category while the denominator calculates the user's overall visits to distinct categories during all seasons:

$$W_{ul} = \frac{\sum_{\forall l_{u_{i,k}} \in l_i} c_{u_i,l}}{\sum_{\forall l_{u_{i,k}} \in L} R_{u_i,L}} \tag{4}$$

Category

Five categories are being selected for POIs: Mountains, Rivers, Lakes, Restaurants, and Cities. Every location l must belong to one of the above categories. That is, we have divided the entire dataset into different categories and then extracted the desired category corresponding to the interest of each user.

Category-location graph

To represent the relationship between a location and category, the proposed model uses a directed bipartite graph which is different from the previous one based on the weighting mechanism adopted. It is represented by $G_{kl} = (K \cup L, W_{kl})$, in which L and K denote the sets of locations and categories, respectively. W_{kl} represents weighted edges and can be computed using Eq. (5). It can be computed as the number of times a place l is visited in a specific category k against all check-ins in the concerned category:

$$W_{kl} = \frac{\sum_{\forall c_l \in k_i} c_l}{\sum_{\forall c_L \in k_i} c_L} \tag{5}$$

Category-user graph

This graph emphasizes the importance of a category that corresponds to each user. The proposed model extracts categories for each POIs using Algorithm 2, which is a modified version of an algorithm proposed in [15]. It correlates each user from U with a category $k \in K$. The graph is denoted by $G_{uk} = (U \cup K, W_{uk})$, where W_{uk} denotes a set of weights between users and categories as computed in Eq. (6), the number of times a user visited the desired category to the total number of visits to all categories made by the same user:

$$W_{ku} = \frac{\sum_{\forall l_{u_{i,k}} \in t_i} c_{u_i,t}}{\sum_{\forall l_{u_{i,k}} \in T} c_{u_i,T}} \tag{6}$$

Category-category graph

It is a bidirectional bipartite graph representing the relationship between pair of categories. For example, if we take two categories viz, k and $k^{'}$ which are linked using a link if a certain user u check-in both in the same season s. Using this intuition, we construct the graph as $G_{kk'} = \left(K \cup K, W_{kk'}\right)$, in which K denotes the set of categories, and $W_{kk'}$ is the weighted edges between pair of categories. The weight between categories is calculated using Eq. (7) as it represents the number of times a user u has visited the corresponding categories simultaneously in season s.

$$W_{kk'} = \frac{\sum_{\forall u_{l_{i,k}} \in s_i} c_{u_i,s}}{\sum_{\forall u_{l_{i,k}} \in S} C_{u_i,S}} \tag{7}$$

Inputs
C: Users' check-in history
ΔS: Season

Output
K: List of categories in seasons

```
i    Sort C-based date wise all check-ins
ii   Generate S time periods with season $\Delta S$
iii  Generate list of categories $K$

iv   for $i \leftarrow 0$ to $i$ equal to $S$   // Every category $k_i \in S$
v        n ← 0
vi       while ($S_{start} \le c_{u_i}[n] \le S_{end}$) do     // While $c[n] \in S_i$
vii          S[n] ← $c_{u_i}[n]$;
viii         n ++
ix       q ← 0
x        for $j \leftarrow 0$ to $lenghtof(S[i])$   // Every category $l \in S_i$
xi           if ($u_i[j].category ==$ K[$j$])       // Attempted by user $u$
xii              K[q].add($c_{u_i}[n]$)
xiii         Else
xiv              q ++
xf   return K
```

Algorithm 2. Extracting most visited categories in each season for all users

Algorithm 2 helps to extract the most visited categories of locations in each season. Such an algorithm aims to get the categories users mostly prefer in the desired season(s). For example, a user u_i may love to visit historical places in summer, whereas in spring, she/he prefers to hike the mountains. Considering this approach to provide the most preferable recommendation in various periods can be helpful. Algorithm 2 accepts check-in history and seasons, respectively. Step 1 is sorting the check-ins based on date, whereas steps 2 and 3 create lists for time (dividing the dataset into seasons) and categories (categories like mountains, rivers, lakes, meadows, cities, and parks). Step 4 goes through for each season, and n is declared in step 5 to keep track of each check-in of users. Step 6 is whether check-ins are fall or not in the desired seasons. Step 7 runs and adds the check-in to the season list if true. Step 8 increment the n, whereas step 9 initializes q to track the desired category in a season. Step 10 runs to make the check-in in the desired category by checking it in step 11. If it is true, the desired check-in is added to the category list in step 12. For false, it increments the q in step 14 and reruns for the next season. Finally, step 15 returns list K.

Category-season graph

A directed bipartite graph represents relations between a season and a category. The category-season graph is denoted by $G_{ks} = (K \cup S, W_{ks})$, in which K and S show the categories and seasons, respectively. W_{ks} is a set of weights established between category and season. Weighted edges between season and category are computed using Eq. (8), where the numerator computes the number of check-ins performed in the category k by all users in seasons s_i, while the denominator represents the whole check-ins performed in all seasons for the same category.

$$W_{ks} = \frac{\sum_{\forall c_U \in s_i} |n_{ks}|}{\sum_{\forall c_U \in S} |n_{ks}|} \tag{8}$$

Location-location graph

This bipartite graph is employed to show the spatial distance between locations if a user u visits two locations l and l' at the same time and distance in a range R_g, then a link is established between them. The graph is denoted as $G_{ll'} = \left(L \cup L, W_{ll'}\right)$, in which L is a set of locations and $W_{ll'}$ is a set of weights among them as computed by Eq. (9) using geographical proximity.

$$W_{ll'} = 1 - \frac{geodist_{ll'}}{R_g} \tag{9}$$

Location-user graph

It represents the relationship between locations and users, also known as a directed bipartite graph. Also, users' interests correspond to a specific location that changes over time. More specifically, this relation is represented as $G_{lu} = (L \cup U, W_{lu})$. The weight W_{lu} is calculated using Eq. (10) as the number of times a user u visit a place l to the total number of check-ins made by all the users to that location:

$$W_{lu} = \frac{\sum_{\forall c_u \in l_i} c_u}{\sum_{\forall c_U \in l_i} c_U} \tag{10}$$

Location-season graph

To show the significance of a certain location for a user u in a season s_i, the proposed model uses a location-season graph, which can be represented by $G_{ls} = (L \cup S, W_{ls})$, L and S show a set of locations and seasons. W_{ls} represents the weighted edges as calculated using Eq. (11).

$$W_{ls} = \frac{\sum_{\forall c_U \in l_i} |n_{ls}|}{\sum_{\forall c_U \in S} |n_{ls}|} \tag{11}$$

Problem definition

Some tourists visit a set of locations in a particular season, while some tend to visit each season. Furthermore, users' preferences change by location category K, *i.e.*, u_i may love to visit l_i in winter whereas u_j in summer. To consider this problem, we must provide a list of locations L to user u in a season s belong to category k_i given that $Q(u, l, s)$.

Proposed next-POI recommender system

This section presents the proposed POI recommendation model that jointly learns multiple graph embeddings and encodes them into a low-dimensional embedding space exploiting semantic relations between the nodes of the networks.

Learning embeddings for large information networks

Two nodes m and n are directly connected by an edge, known as first-order proximity. In contrast, the relation between vertices that share multiple neighbor nodes but are not directly associated with each other is referred to as second-order proximity. The LINE model [23] tries to learn the relationships of large graphs to extract this kind of proximity. With this concept, we expand our model to learn the embeddings of large information networks.

Consider two disjoint sets $= (Q_A \cup Q_B, W)$, where the vertices in Q_A that collaborate many common neighbors with Q_B but they are not linked, then there is a high probability that their distributions are the same. To compute the conditional probability of vertex $n_j \in Q_B$ given node$m_i \in Q_A$, the model employs the following equation:

$$p\left(n_j|m_i\right) = \frac{exp(\overrightarrow{n}_j^T \times m_i)}{\sum_{u_n \in Q_B} exp(\overrightarrow{n}_n^T \times \overrightarrow{m}_n)} \tag{12}$$

The vectors of m_i and n_j can be represented as $\overrightarrow{m}_i$ and $\overrightarrow{n}_j$. Hence, for each vertex $m_i \in Q_A$, Eq. (12) presents conditional distribution $p(\bullet|m_i)$ to all related vertices in the set Q_B. Then, the model uses the conditional distribution to approximate the empirical distribution $\widehat{p}(\cdot|m_i) = \frac{w_{i,j}}{\sum w_{i,m}}$ employing the following objective function:

$$O = \sum_{u_i \in Q_A} \lambda_i d(\widehat{p}(\bullet|m_i), p(\cdot|m_i)) \tag{13}$$

where $d(\bullet|\bullet)$ indicates the Fullback–Leibler divergence between conditional and empirical distributions. To tune the significance of m_i, we have used λ_i as a hyper-parameter. This parameter is set to the outdegree of each node. Thus, Eq. (13) tries to optimize the following objective function:

$$O = -\sum_{e_{i,j} \in W} w_{i,j} logp\left(n_j|m_i\right) \tag{14}$$

The $\left\{\overrightarrow{\boldsymbol{m}}_{\boldsymbol{i}}\right\}_{i=1...\boldsymbol{Q_A}}$ and $\left\{\overrightarrow{\boldsymbol{n}}_{\boldsymbol{j}}\right\}_{\boldsymbol{j}=1...\boldsymbol{Q_B}}$ that minimizes Eq. (14) are the low-dimensional nodes representations in $\mathbb{R}^d$ [15].

Optimization of the model

It requires the summation of the complete set to find conditional probability as a result, it enhances computational complexity. To address this problem, we use negative sampling used in [27], which simply samples N negative edges according to the noise distribution for every edge *(i, j)* as defined in the following equation.

$$log\sigma\left(\vec{n}_j^T \times \vec{m}_i\right) + \sum_{h=1}^{N} W_{u_n} \sim P_n(n)\left[log\sigma - \left(\vec{n}_j^T \times \vec{m}_i\right)\right] \tag{15}$$

where $\sigma(x) = 1/1 + exp(-x)$ is the sigmoid function, and $P_n = \left(n \propto d_h^{3/4}\right)$ same as proposed in [29], d_h is the out-degree of node n. Furthermore, we come up with an asynchronous stochastic gradient algorithm [33] to optimize Eq. (15). If an edge (i, j) has been sampled, the gradient concerning to the embedding of $\overrightarrow{m}_i$ of node i can be computed as:

$$\frac{\partial O}{\partial \overrightarrow{m}_i} = w_{i,j} \times \frac{\partial logp\left(n_j|m_i\right)}{\partial \overrightarrow{m}_i} \tag{16}$$

It is also considered that the gradient is multiplied by the weight of a link in Eq. (16). It may be problematic if we ignore the balancing of the learning rate. We should have to carefully keep the learning rate because, when selecting the learning rate according to the links with low weights, the gradients on links with high weights will be disastrous. Similarly, when selecting the learning rate with high weight, the gradient will be too small. The model employs the sampling approach adopted in [31] to sample a random edge. Finally, the model draws a sampled edge using alias table according to [29], which minimizes computational complexity to $O(1)$. Table 3 illustrates the complexity of edge sampling optimization process.

Learning graph dynamics

Initially, by providing bipartite input graphs, the subsequent step combines them into the model. Our input

Table 3 Net Complexity of sampling

Sample edge	$O(1)$
Optimization of negative sampling	$O(N + 1)$
Total complexity	$O(N \times E)$

graphs have been divided into three parts, one considering user networks (User-Location, User-Season, User-Category, User-User). At the same time, the second is related to location (Location-Location, Location-User, Location-Season), and the third one corresponds to a category of places (Category-User, Category-Category, Category-Location). The model collectively integrates the embedding of participating graphs as defined in Eq. (18–28) related to users and POIs relations, optimizing the objective function defined in Eq. (17) as follows:

$$\begin{aligned} O = O_{ul} + O_{us} + O_{uk} + O_{uu} + O_{ll} + O_{lu} \\ + O_{ls} + O_{ku} + O_{kk} + O_{kl} + O_{ks} \end{aligned} \tag{17}$$

The model computes these objective functions as follows.

$$O_{ul} = -\sum_{e_{i,j} \in W_{ul}} w_{i,j} logp\left(u_i|l_j\right) \tag{18}$$

$$O_{us} = -\sum_{e_{i,j} \in W_{us}} w_{i,j} logp\left(u_i|s_j\right) \tag{19}$$

$$O_{uk} = -\sum_{e_{i,j} \in W_{uk}} w_{i,j} logp\left(u_i|k_j\right) \tag{20}$$

$$O_{uu} = -\sum_{e_{i,j} \in W_{uu}} w_{i,j} logp\left(u_i|u_j\right) \tag{21}$$

$$O_{ll} = -\sum_{e_{i,j} \in W_{ll}} w_{i,j} logp\left(l_i|l_j\right) \tag{22}$$

$$O_{lu} = -\sum_{e_{i,j} \in W_{lu}} w_{i,j} logp\left(l_i|u_j\right) \tag{23}$$

$$O_{ls} = -\sum_{e_{i,j} \in W_{ls}} w_{i,j} logp\left(l_i|s_j\right) \tag{24}$$

$$O_{ku} = -\sum_{e_{i,j} \in W_{ku}} w_{i,j} logp\left(k_i|u_j\right) \tag{25}$$

$$O_{kk} = -\sum_{e_{i,j} \in W_{kk}} w_{i,j} logp(k_i|k) \tag{26}$$

$$O_{kl} = -\sum_{e_{i,j} \in W_{kl}} w_{i,j} logp\left(k_i|l_j\right) \tag{27}$$

$$O_{ks} = -\sum_{e_{i,j} \in W_{ks}} w_{i,j} logp\left(k_i|s_j\right) \tag{28}$$

For optimization of object functions, as defined in Eq. (17), it requires merging the links of all networks, and at every step, the model updates a new sample edge. The probability of the desired link is computed based on its associated weight. The training of our model is done jointly using the algorithm in [15] dynamically.

Personalized next-POI recommendation

After exploiting semantic relations between the nodes of the participating graphs and learning their embeddings, the next step is to make recommendations for a user. Given a query $Q(u, l, s)$, which specifies a user u in a location l and season s we can correspond these values to desire season s. Claiming that a user is willing to attend locations in specific seasons corresponding to category k_i. Finally, we rank a list with *top@n* unvisited location for a user u_i related to category k_i. The proposed model uses Eq. (29) to recommend unvisited locations:

$$Q(u, l, s) = \alpha \times \left(\vec{u}^T \times \vec{l}\right) + \beta \times \left(\vec{k}^T \times \vec{l}\right) + \gamma \times \left(s^T \times \vec{l}\right) \tag{29}$$

where $\vec{u}$ and $\vec{l}$ is the vector representations of user u, and location l. Similarly, $\vec{k}$ represents the vector representation of the category k such check-in. The proposed model uses cloud computing to store and process the data, and to jointly learn the vector representations from various information graphs in the same embedding space. This allows for more efficient handling of large amounts of data and the ability to perform complex computations. The cloud-based solution also allows for the incorporation of more information networks, which in turn reduces sparsity by incorporating more information networks, and jointly learns the dynamics of the *social influence* ($\vec{u}^T \times \vec{l}$), the *geographical influence* ($\vec{k}^T \times \vec{l}$), and the *temporal influence* ($\vec{s}^T \times \vec{l}$), simultaneously to provide more accurate and personalized POI recommendations to users.

Employment of cloud and edge computing

Edge computing is a distributed computing paradigm that focuses on processing data near the source of data generation, thereby reducing latency and bandwidth usage. The proposed work leverages edge computing to process data generated by tourists using LBSNs to share their preferences and interests. LBSNs enable the collection of tourists' location and interest data, which can be processed at the network's edge, facilitating real-time and personalized recommendations. In addition to edge computing, cloud computing can be employed to store and

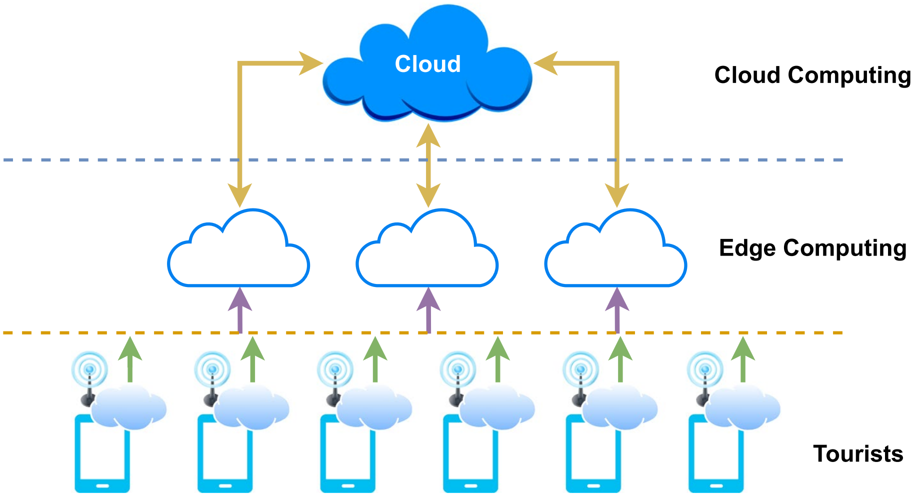

Fig. 2 Use of edge and cloud computing in the proposed model

process the large amounts of data collected from LBSNs as shown in Fig. 2 consists of two main components. The first component shows tourists using LBSNs to share their preferences and interests, generating data collected and processed at the network's edge through edge computing. The second component displays the cloud computing infrastructure used to store and process the large amounts of data collected from LBSNs. This approach enhances the computational capabilities of the proposed model and enables tourism management systems worldwide to access and use it easily.

Results and discussion

This section focuses on the performance evaluation of the proposed model based on the Foursquare dataset. That is, we compare the results of the proposed model with existing POI models.

Dataset

To analyze the results of the proposed model, we have used a publicly available dataset known as Foursquare.[1] The dataset consists of POIs, users' check-ins, and friendships, which have been collected from the year 2012 to 2014. The distribution of the dataset is depicted in Table 4. The seasons are extracted using the period given in the dataset. Similarly, each POI is associated with its category like restaurant, river, lake, city, and so on.

Table 4 Distribution of dataset

POIs	1048000
Check-ins	2145800
Friendships	607300

Baseline models

In this set of experiments, we have comparatively viewed the results with the following models.

- **RELINE** [10]: They have used a graph-based approach to learn users' and POI relationships from 8 weighted networks in a hidden space and provide location recommendations under a strategy having a probability that examines the influence of social, geographical, temporal, and preference dynamics.
- **GE** [28]: is another graph-based embedding model that exploits geographical influence, sequential effect, temporal cyclic effect, and semantic effect in a unified way and embeds four information graphs into a shared embedding space. Also, a novel time-decay method is proposed that dynamically computes the user's latest preferences based on the embedding of his/her checked-in POIs learned in the embedding space.
- **WWO** [31]: is a unified POI recommender system with temporal interval assessment that considers temporal interval distributions and developed the low-rank network model, identifying a set of bi-weighted network bases to learn the static preferences and dynamic preferences coherently.
- **PGB** [33]: This probabilistic model employs the graph-based Markov chain method to improve recommendation accuracy. The choice of suggesting an item is conditioned by considering recommendations generated in previous steps.

[1] https://sites.google.com/site/yangdingqi/home/foursquare-dataset?pli=1

Experiments evaluation

To conduct experiments, we have divided the check-ins into three partitions for each target user that include: (i) the training set ε_C^T is the known information, which comprises 80% of the entire check-ins, (ii) the probe set ε_C^P, used to test the model, and it contains 10% of the data, and (iii) the validation set ε_C^V is the rest 10% for validation of the proposed model. Mathematically we can represent it as $\varepsilon_c = \varepsilon_C^T \cup \varepsilon_C^P \cup \varepsilon_C^V$ and $\varepsilon_C^T \cap \varepsilon_C^P \cap \varepsilon_C^V = \varnothing$. To make recommendations for a target user, the model uses the POIs in set ε_C^T.

For the evaluation, we measure the *Accuracy@n* as proposed in [15]. For each $l \in \varepsilon_C^P$ given as a query $Q(u, l, s)$, the prediction score for that specific location l along with all unvisited proximate POIs of the target user is computed based on Eq. (30). The model ranks the predicted scores into a list, and then chooses the top POIs. If the ground truth $l \in \varepsilon_C^P$ exists in the top recommendations, then the model has accurately recommended that location (i.e., True Positive), otherwise it has suggested a wrong location. To calculate the net accuracy of ten recommendations, the model averages all predictions test cases as follows:

$$Accuracy@n = \frac{\#TruePositive@n}{\varepsilon_C^P} \tag{30}$$

Impact of information graphs

Particularly, we examine how the integration of an information graph influences the top-n predictions. Thus, we compare NPR-LBN with the four models PGB, GE, WWO, and RELINE which are described in Baseline models section. The results shown in Table 5 reveal that as the model integrates the latest information network, its accuracy increases. This way, the proposed model lessens the sparsity issue by exploiting rich information about the users or the POIs. Also, it is noticeable that the accuracy of all models increases with n, which exhibits that the models fit well to users' behavior.

Comparative analysis

This section evaluates the results of the proposed work with baselines using accuracy to produce top@n recommendations [$n = 1, 5, 10, 15, 20$] employing a real-world dataset (Foursquare). Specifically, judging the performance of all models that provide POI recommendations against cold-start users and locations.

Table 5 Impact of additional information networks to the model

Model	1	5	10	15	20
PGB	0.21	0.24	0.32	0.33	0.43
GE	0.25	0.32	0.37	0.40	0.42
WWO	0.22	0.28	0.34	0.34	0.44
RELINE	0.28	0.32	0.37	0.44	0.46
NPR-LBN	**0.29**	**0.31**	**0.38**	**0.45**	**0.49**

Cold start user

We have studied the efficacy of the proposed work considering the cold-start user problem. Producing recommendations for such kinds of users is incredibly challenging due to the unavailability of the required information. In this context, we conducted experiments to provide recommendations only to cold-start users and used an accuracy metric to analyze the results of these models, as shown in Fig. 3. Since all models mentioned above support the cold-start recommendation, in this regards we compare our model with all these baselines. We employ side information related to users and locations from eleven information graphs, yielding improved results.

Cold start locations

Similarly, we examine a problem that is known as cold-start locations. The aim is to suggest locations that were not visited for at least one or few check-ins at a location with less than 25 check-ins. Thus, to cope with this, we have investigated how models behave on unpracticed users or new location is introduced into the system. In addition, we have analyzed whether the new location is listed in the top@n recommendations. Again, the proposed model outperforms in terms of producing quality recommendations, as shown in Fig. 4.

Significance of seasons

Here, we have analyzed the influence of the time interval per season ΔS against the accuracy for different values of the ten recommendations made by the model. ΔS significantly impacts the model's results since it is employed to build multiple information graphs. Table 6 shows the results of the model using the dataset. We can notice that the accuracy score reaches a maximum value at a certain point and then decreases gradually. The reason behind the low accuracy score is the value of ΔS. If its value is exceedingly small, then it means we have less data, and diffidently the accuracy score will be less. On the contrary, for large values of ΔS, a substantial number of nodes related to the target user exists, which causes an overfitting problem. Considering these factors, we choose the size value for the dataset to be 15.

Conclusion and future work

Due to the exponential growth of information on the internet, recommender systems have become prevalent technological assistants to humans. Likewise, LBS have

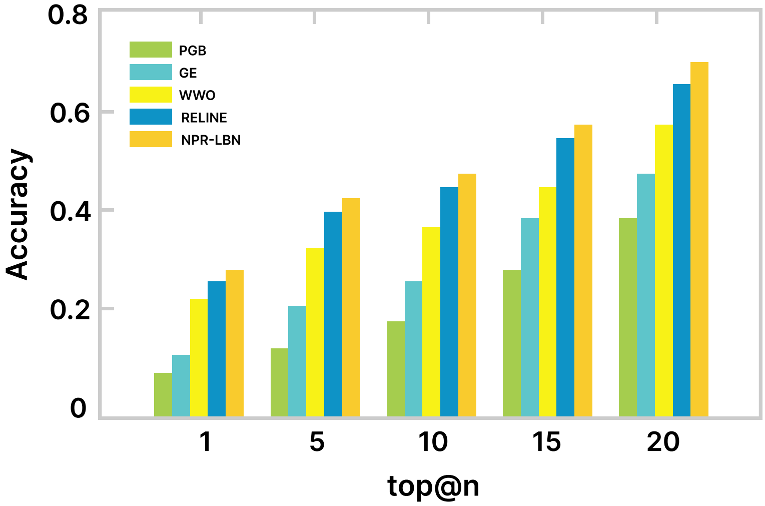

Fig. 3 Accuracy for cold-start users

become ubiquitous in different sectors and therefore gained the attention of numerous scientific disciplines. The emergence and usage of communication technologies such as mobile devices allow researchers to come up with more elegant solutions against sparsity and cold-start problems. Geographical information shared on such networks enables researchers to tackle both problems. Various POI approaches that have been proposed using social influences, geographical proximity, and preference dynamics using their social influences. In this work, we have considered cold-start and sparsity problems while providing POI recommendations in the tourism sector. Particularly, our proposed model has been upgraded and comes with additional features such as users' satisfiability on the system and employs a weighted probabilistic approach over eleven information graphs based on relations established among users, seasons, and categories. The model personalizes locations by jointly learning the embeddings of users and POIs into the same embedding space. The incorporation of edge and cloud computing in our proposed model has improved the system's accuracy and scalability, allowing it to be easily used by tourism management systems worldwide. The influence of social, geographical, and temporal factors in terms of accuracy has been scrutinized. Our model has been evaluated and outperformed against the cold-start users and the cold-start locations. Our future work is to ensure users' social information security, which is crucial to users, and a state-of-the-art problem in POI recommendation.

Table 6 Impact of time period/seasons ΔS over accuracy for top@n recommendations

ΔS	Acc@n				
5	0.27	0.35	0.24	0.25	0.32
10	0.23	0.32	0.42	0.31	0.32
15	**0.32**	**0.38**	**0.39**	**0.42**	**0.45**
20	0.33	0.42	0.23	0.98	0.32

Acknowledgement

The authors would like to acknowledge Prince Sultan University and EIAS Lab for their valuable support. Further, the authors would like to acknowledge Prince Sultan University for paying the Article Processing Charges (APC) of this publication.

Conflicts of interest

All the authors have no conflicts of interest.

Authors' contributions

I.K and A.S are the main writers of the manuscript. They put forward the main ideas of architectural modeling and analysis, and wrote the main part of this manuscript. G.A and M.A supervised the work and provided the funding for the said research. T.S have designed the Algorithms and R.K and validated the results. All the authors reviewed and approved this manuscript.

Availability of data and materials

The materials that support this study are available upon request from the second author.

Declarations

Ethics approval and consent to participate

The study did not require ethical approval.

Competing interests

The authors declare no competing interests.

Received: 25 November 2022 Accepted: 18 March 2023
Published online: 11 April 2023

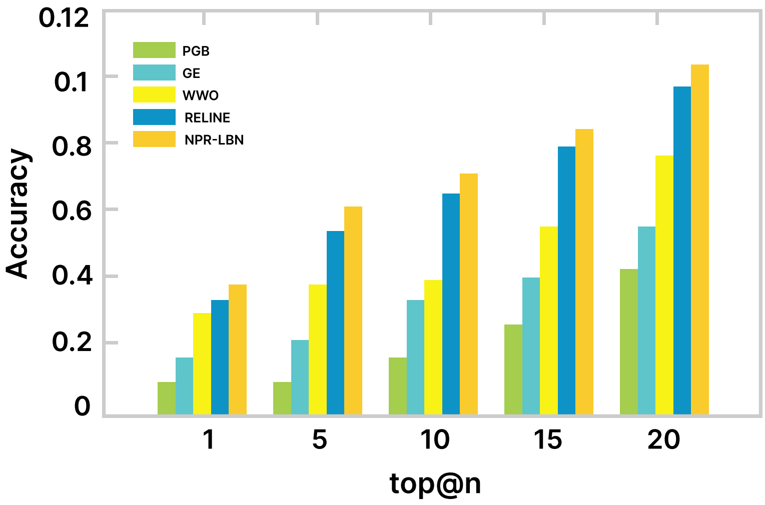

Fig. 4 Accuracy for cold-start POIs

References

1. Hamid RA, Albahri AS, Alwan JK, Al-qaysi ZT, Albahri OS et al (2021) How smart is e-tourism? A systematic review of smart tourism recommendation system applying data management. Comput Sci Rev 39:100337
2. Ashley C, Brine PD, Lehr A, Wilde H (2007) Introduction. The role of the tourism sector in expanding economic opportunity, 1st ed, Coporate Social Resoponsibility Initiative Report No. 23, Cambridge, England. Kennedy School of Government Harvard University, MA, pp 6–7
3. Manzoor F, Wei L, Asif M, Haq MZ, Rehman H (2019) The contribution of sustainable tourism to economic growth and employment in Pakistan. Int J Environ Res Public Health 16(19):3785
4. Pei Y, Zhang Y (2021) A study on the integrated development of artificial intelligence and tourism from the perspective of smart tourism. J Phys: Conf Ser 1852(3):2021
5. Farooqi MM, Shah MA, Wahid A, Akhunzada A, Khan F, Ali I (2019) Big data in healthcare: a survey. Applications of intelligent technologies in healthcare. Springer, Cham, pp 143–152
6. Malik A, Khan MZ, Faisal M, Khan F, Seo JT (2022) An efficient dynamic solution for the detection and prevention of black hole attack in VANETs. Sensors 22(5):1897
7. Abbas S, Talib MA, Ahmed A, Khan F, Ahmad S, Kim DH (2021) Blockchain-based authentication in internet of vehicles: a survey. Sensors 21(23):7927
8. Hoq KMG (2016) Information Overload: Causes, Consequences and Remedies - A Study. Philosophy and Progress 55(1-2):49–68. https://doi.org/10.3329/pp.v55i1-2.26390
9. Achmad KA, Nugroho LE, Djunaedi A (2017) Tourism contextual information for recommender system. In 2017 7th International Annual Engineering Seminar (InAES) (pp. 1-6). IEEE
10. Christoforidis G, Kefalas P, Papadopoulos AN, Manopolous Y (2021) RELINE: point-of-interest recommendations using multiple network embeddings. Knowl Inf Syst 63:791–817
11. Bobadilla J, Ortega F, Hernando A, Gutiérrez A (2013) Recommender systems survey. Knowl-Based Syst 46:109–131
12. Mahadik K, Wu Q, Li S, Sabne A (2020) Fast distributed bandits for online recommendation systems. In Proceedings of the 34th ACM international conference on supercomputing (pp. 1-13)
13. Lu J, Wu D, Mao M, Wang W, Zhang G (2015) Recommender system application developments: a survey. Decis Support Syst 74:12–32
14. Linden G, Smith B, York J (2003) Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing. 7(1):76–80
15. Figueredo M, Ribeiro J, Cacho N, Thome A, Cacho A, Lopes F, Araujo V (2018) From photos to travel itinerary: A tourism recommender system for smart tourism destination. In 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService) (pp. 85-92). IEEE
16. Bahramianm Z, Abbaspour RA, Claramunt C (2017) A cold start context-aware recommender system for tour planning using artificial neural network and case-based reasoning. Mob Inf Syst 2017:18
17. Abbas A, Zhang L, Khan SU (2015) A survey on context-aware recommender systems based on computational intelligence techniques. Computing 97:667–690
18. Artemenko O, Kunanets O, Pasichnyk V (2017) E-tourism recommender systems: a survey and development perspectives. Int Q J 6(2):91–95
19. Bahramian Z, Abbaspour RA, Claramunt C (2017) A CONTEXT-AWARE TOURISM RECOMMENDER SYSTEM BASED ON A SPREADING ACTIVATION METHOD. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 42. Tehran, pp 7–10
20. Levandoski JJ, Sarwat M, Eldawy A, Mokbel MF (2012) Lars: A location-aware recommender system. In 2012 IEEE 28th international conference on data engineering (pp. 450-461). IEEE
21. Thasal R, Yelkar S, Tare A, Gaikwad S (2018) Information retrieval and de-duplication for tourism recommender system. Int Res J Eng Technol 5(3):1683–1687
22. Paulino I, Lozano S, Prats L (2021) Identifying tourism destinations from tourists' travel patterns. J Destin Mark Manag 19:100508
23. Vijayakumar V, Vairavasundaram S, Logesh R, Sivapathi A (2019) Effective knowledge-based recommender system for tailored multiple point of interest recommendation. Int J Web Portals (IJWP) 11(1):1–18
24. Khan F, Ahmad S, Güruler H, Cetin G, Whangbo T, Kim CG (2021) An Efficient and Reliable Algorithm for Wireless Sensor Network. Sensors 21(24):8355
25. Khan F, Gul T, Ali S, Rashid A, Shah D, Khan S (2018) Energy aware cluster-head selection for improving network life time in wireless sensor network. In Science and Information Conference. Springer, Cham, pp 581–93
26. Xie M, Yin H, Wang H, Xu F, Chen W, Wang S (2016) Learning graph-based poi embedding for location-based recommendation. In Proceedings of the 25th ACM international on conference on information and knowledge management (pp. 15-24)
27. Xie M, Yin H, Xu F, Wang H, Zhou X (2016) Graph-based metric embedding for next poi recommendation. In Web Information Systems Engineering–WISE 2016: 17th International Conference, Shanghai, China, November 8-10, 2016, Proceedings, Part II 17 (pp. 207-222). Springer International Publishing
28. Liu Y, Liu C, Liu B, Qu M, Xiong H (2016) Unified point-of-interest recommendation with temporal interval assessment. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1015-1024)
29. Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q (2015) Line: Large-scale information network embedding. In Proceedings of the 24th international conference on world wide web (pp. 1067-1077)
30. Kenteris M, Gavalas D, Mpitziopoulos A (2010) A mobile tourism recommender system. In The IEEE symposium on Computers and Communications (pp. 840-845). IEEE

31. Ding F, Zhu G, Li Y, Zhang X, Atrey PK et al (2021) Anti-forensics for face swapping videos via adversarial training. IEEE Trans Multimedia 24:3429–3441
32. Leskovec J, Faloutsos C (2006) Sampling from large graphs. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 631-636)
33. Joorabloo N, Jalili M, Ren Y (2019) A probabilistic graph-based method to improve recommender system accuracy. In Engineering Applications of Neural Networks: 20th International Conference, EANN 2019, Xersonisos, Crete, Greece, May 24-26, 2019, Proceedings 20 (pp. 151-163). Springer International Publishing

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.