This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access

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

Digital Object Identifier 10.1109/ACCESS.2021.DOI

SgWalk: Location Recommendation by User Subgraph Based Graph Embedding

DENIZ CANTURK[1], PINAR KARAGOZ[2](Member, IEEE)

[1]Middle East Technical University (METU), Department of Computer Engineering, Ankara, 06800, Turkey (e-mail: deniz.canturk@metu.edu.tr)
[2]Middle East Technical University (METU), Department of Computer Engineering, Ankara, 06800, Turkey (e-mail: karagoz@metu.edu.tr)

Corresponding author: Deniz Cantürk (e-mail: deniz.canturk@metu.edu.tr).

ABSTRACT Popularity of Location-based Social Networks (LBSNs) provide opportunity to collect massive multi modal datasets that contain geographical information, as well as time and social interactions. Such data is a useful resource for generating personalized location recommendations. Such heterogeneous data can be further extended with notions of trust between users, popularity of locations and expertise of users. Recently the use Heterogeneous Information Network (HIN) models and graph neural architectures have proven successful for recommendation problem. One limitation of such solution is on capturing the contextual relationships between the nodes in the heterogeneous network. In location recommendation, spatial context is a frequently used consideration such that users prefer to get recommendations within their spatial vicinity. To solve this challenging problem, we propose a novel Heterogeneous Information Network (HIN) embedding technique, SgWalk, which explores the proximity between users and locations and generates location recommendations via subgraph-based node embedding. SgWalk follows four steps: building users subgraphs according to location context, generating random walk sequences over user subgraphs, learning embeddings of nodes in LBSN graph, and generating location recommendations using vector representation of the nodes. SgWalk is differentiated from existing techniques relying on meta-path or bi-partite graphs by means of utilizing the contextual user subgraph. By this way, it is aimed to capture contextual relationships among heterogeneous nodes more effectively. The recommendation accuracy of SgWalk is analyzed through extensive experiments conducted on benchmark datasets in terms of top-n location recommendations. The accuracy evaluation results indicate minimum 23% (@5 recommendation) average improvement in accuracy compared to baseline techniques and the state-of-the-art heterogeneous graph embedding techniques in the literature.

INDEX TERMS Location-based social networks, Location recommendation, Heterogeneous Graph Embedding, Information Fusion, Trust-aware recommendation

I. INTRODUCTION

The rapid growth of the mobile technologies led to development of online social networks with an increasing user counts. As reported in [2], it is expected that it will reach to 4.2 billion social media users by the end of 2021. Additionally, the advancements in positioning technology allowed utilizing location data through the social networks in various ways, such as sharing an instant location with other users or exchanging travel experiences with the friends [52]. These opportunities foster sharp increases in use of LBSNs (Location-Based Social Network), such as Foursquare. An LBSN stores users' check-in history, which is a precious resource to analyze spatio-temporal user behaviour and to generate recommendations for a variety of aspects such as friends, locations, and activities according to the context of the user [28]. According to [1], data sets provided by Foursquare is used by more than 125,000 developers worldwide. Given an LBSN check-in history (as a set of tuples consisting of user, check-in time, check-in location information) and the user's social network connections, location recommendation for LBSN aims to recommend a set of locations for the target user. As there are too many users and locations in an LBSN, processing and analyzing a formed social networks requires considerable effort.

To overcome this difficulty, the use of graph structures and network embedding have gained popularity in recent years with the idea of representing vertices (nodes) in a graph using a low-dimensional vector space that preserves the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access

fundamental structural properties of the graph. Hence, network embedding facilitates identifying nodes having closer characteristics to each other according to their corresponding vector representations. This leads to the adoption of network embedding approaches in various graph analysis tasks by considering node distances (similarities). Recent graph embedding techniques can be classified into two categories depending on the types of input graphs, i.e., homogeneous or heterogeneous graph embedding [14]. In a homogeneous graph, which has nodes of single type only (such as users in a social network), graph embedding techniques use sequences of nodes obtained via randomly sampled from the graph (using random walks over the graph [38]) for preserving the proximity between node pairs. On the other hand, a heterogeneous graph has a complex structure containing nodes of different types representing different entities, and edges linking nodes of the same type or across different entities [16], [19]. For example, an LBSN graph with user (U), location (L), and friend (F) entities contains not only the homogeneous edges between users (friendship) but also heterogeneous edges linking a user and location (visits). These types of graphs are also known as heterogeneous information networks (HINs). Therefore, HINs embody more information, contain richer semantics compared to the homogeneous graphs, and facilitates more complex analysis on heterogeneous data [33].

A heterogeneous graph can be directly embedded by homogeneous graph embedding techniques. Still, it suffers from poor performance in results [14] due to considering different types of nodes as if they are the same. Therefore, network embedding techniques for heterogeneous graphs need to consider information fusing and structure of the graph while generating node sequences. Many researchers have been working on the techniques for HIN embedding [16], [19], [25], [41], [44], [45], [49], [53], and have proved the usefulness of HIN embedding in HIN analysis tasks. In such studies, some of the techniques rely on meta-paths in a random walk to generate node sequences for preserving the similarity between nodes [16], [19], [41]. Some other techniques decompose the HIN into simple networks (bi-partite graphs) and then jointly learn the embeddings of each sub-network [13], [44], [53]. Additionally, there are solutions that employ neural network-based approaches to learn non-linear mapping functions for HIN embedding [25], [49], [50], [56]. HIN's embedding learned by the aforementioned techniques have been used in many tasks, such as classification [26], clustering [30], link prediction [31], and recommendation [33], [59].

One important limitation in the existing HIN embedding methods is that they lack a clear mechanism to cope with spatial context in recommendation. In location recommendation, spatial context is a frequently used consideration such that users prefer to get recommendations within their spatial vicinity. Therefore it is needed to capture the complex relations between different types of entities locally with respect to the spatial regions (spatial contexts). To be able to cope with this problem, in this paper, we propose a new heterogeneous graph embedding technique SgWalk (***S***ub***g***raph ***Walk***), that decomposes the graph into smaller subgraphs constructed for a user with respect to spatial context of the user, which is considered as the vicinity of the user (the region with a given radius around the user's current position). In the constructed user subgraphs, we consider a wide range of information sources such as personal, social, spatial, and trustworthiness contexts. Inclusion of trust relationship is also novel for constructing heterogeneous networks for LBSNs [52]. Location context based user subgraph, which is a HIN including a variety of entities and the node embeddings constructed by the proposed method are utilized in node to generate a more accurate recommendations for the given user. The SgWalk technique consists of a four-step process that explores the proximity between users and locations and generates recommendations. In the first step, it builds user subgraph according to the spatial contexts and as the second graph it generates random walk sequences over the user subgraphs. In the third step, it learns embeddings of nodes in LBSN graph, and finally generates location recommendations for the given user.

The major differences of the proposed node embedding technique are the utilization of contextual user subgraphs instead of bi-partite subgraphs proposed in the literature, and not depending on meta-paths. We can highlight the contributions of the study as follows:

- We propose a user subgraph based graph embedding technique SgWalk. Instead of global LBSN graph, we use subgraphs to capture the relations among heterogeneous entities within spatial contexts.
- We adopt the popular word embedding technique by applying the random walk sampling process on the user subgraph as a new setting for heterogeneous information network embeddings.
- We work on heterogeneous information networks that consist of trustworthiness, user preferences, social connections, and spatial-context notions and fuse them effectively in learning node embeddings. Modeling trustworthiness as trusted user is also a novelty of the study. We analyze the effect of trusted user nodes on the location recommendation performance through ablation studies.
- In order to analyze the recommendation accuracy of the proposed technique, we conduct extensive experiments on four real-world datasets.

The rest of the paper is organized as follows: Section II reviews the related studies in location recommendation and graph embedding. We present preliminaries and the details of SgWalk in Section III. The experiments and the analysis results are given and discussed in Section IV. The paper is concluded in Section V with final remarks.

II. RELATED WORK

In this section, we give a brief overview of previous work addressing recommendation in location recommendation and network embedding techniques.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access

A. LOCATION RECOMMENDATION

Early studies on location recommendation analyzed GPS trajectory logs collected from users of social networking services [12], [60], [61]. However, recent studies employ check-in data in LBSNs due to its availability [5], [11], [51], [55]. There are five major information categories basically used in location recommendation studies: (i) user preferences, to recommend attractive locations matching up attributes of user profile, (ii) social relationships, generating recommendations from the check-in histories of friends who are expected to have similar interests, (iii) spatial-context, clustering locations according to their geographic positions to improve recommendation accuracy, (iv) the temporal effect, utilizing the idea that human activity has time periodicity such as daily or weekly, (v) trust-awareness, relying on the implicit or explicit trust networks of the users.

There is a rich literature considering these five categories in the recommender systems. Zheng et al. [60] converted GPS trajectory data to factorized location-activity matrix to recommend locations for a given activity, or vice versa. Cho et al. [12] investigated human mobility patterns on location-based social networks through temporally and geographically periodic movements with the social network structure. Montjoye et al. [15] showed that human activity has a substantial time periodicity by confirming that users' check-in patterns may be cycled at one day or one week. Gao et al. [21] studied the temporal cyclic patterns of user check-ins in terms of temporal consecutiveness. Temporal cyclic patterns are also employed by Wang et al. [51] together with user's check-in time and the corresponding check-in location to improve the recommendation accuracy.

Focusing on spatial information, Lian et al. [27] proposed a weighted matrix factorization based location recommendation approach integration geographical influence into matrix factorization by using users' and locations' latent feature vectors. Wang et al. [49] introduced a Collaborative Deep Learning (CDL) approach, which extracts deep feature representations from the content and captures the similarity and implicit relationship between locations (and users) by performing deep learning collaboratively. The learned representations are used for location recommendation. Yin et al. [56] improved [49] by adding geographical information and proposed a Spatial-Aware Hierarchical Collaborative Deep Learning model (SH-CDL). This model jointly performs deep representation learning for locations from heterogeneous features and hierarchically additive representation learning for spatial-aware personal preferences.

Since trust concept is one of the entities considered in the graph structure used in our study, we present a quick glance on how trust is used in recommendation studies. Trust is one of the recent and popular concepts employed in graph related problems and recommendation methods for improving accuracy. In a social network, when trust between the users is explicitly available, this information is modeled as an unsigned trust network. When explicit distrust is provided as well as explicit trust, such relationships are considered as a signed trust network. Additionally there are efforts to model implicit trust in order to improve inference on social network [35]. Trust notion has been studied within the scope of a variety of tasks such as influence diffusion modeling [48], and answer selection [43]. For recommendation problems, trust modeling and trust networks are generally used for alleviating the matrix sparsity though auxiliary trust information [3], [4], [54]. To the best of our knowledge, trust modeling has not been incorporated into graph structure for location recommendation in the previous studies.

The summary of location recommendation methods given in this section presents an overview of the employed techniques and how they evolved from matrix factorization towards deep neural architectures. As seen in the summaries, although they focus on the same problem, the settings considered in these studies are different than our approach. One important observation is that the trust concept is not well studied in location recommendation and considering its use in the other recommendation tasks, it promises potential for improvement for location recommendation.

B. NETWORK EMBEDDING

Network embedding aims to find low-dimensional vector representation of the network or its components by preserving the network structure and properties of the nodes in the network [7], [14], [58]. In the early studies [36], [40], dimension reduction techniques decomposed the network for learning the latent low-dimensional vector representations of nodes and edges.

Lately, deep learning-based models take place in network embedding to handle the decomposition issues in large networks. Motivated from Word2Vec [34] approach, Perozzi et al. [38] proposed Deepwalk that fed the generated random walk sequences to the Skip-gram model to learn network representations. DeepWalk assumed random walk sequences as sentences and nodes as words and then maximized the co-occurrence probability of each node pair. Next, node2vec is proposed by Grover et al. [22], in which a biased random walk is performed on homogeneous networks by employing breadth-first and depth-first sampling. With the LINE model, Tang et al. [45] employed both first and second-order proximities between nodes by edge-sampling for learning vectorial representations of nodes in large information networks. Furthermore, higher-order graph proximity for network representations is put forward in the GraRep model by Cao et al. [9]. Unfortunately, these state-of-the-art methods concentrated on learning the representation of nodes in a homogeneous network and cannot be directly applied to heterogeneous networks.

Recently, heterogeneous information networks have gained popularity due to its capability to model nodes representing several different entities and their specific relations. HIN embedding aims to represent all different types of nodes into the shared low-dimensional space. Actually, HIN embedding methods improved the ideas in homogeneous graph embedding research to handle the heterogeneity of the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access

information graph. For example, Metapath2vec [16] as an extension of Deepwalk, performs meta-path based random walks to generate node sequences for accommodating heterogeneous neighborhoods of a node and apply the Skip-gram model to learn the representation of heterogeneous networks. HIN2Vec [19] discovers different types of relationships among nodes by combining meta-paths shorter than a certain length and employs them to generate better walk sequences. JUST [24] performs random walks on the HIN by applying jump and stay strategy to control the walks over the heterogeneous nodes without depending on a meta-path. On the other hand, PTE [44] extends LINE by decomposing the heterogeneous graphs into bi-partite subgraphs and performs individual network embedding by using the LINE approach. GE [53] and JLGE [13] also extend the LINE approach, but for embedding location nodes. Both GE and JLGE decompose LBSN into bi-partite user-location subgraphs and then jointly perform representation learning on node pairs in subgraphs.

Lastly, deep neural network-based methods are imported to heterogeneous data embedding methods due to the power of deep models, such as autoencoder, and graph neural network. SHINE [50] borrows the autoencoder model to encode and decode the heterogeneous information in the social network to obtain the feature representation, and also in [37], [56], the stacked denoising autoencoder (SDAE) is used as the deep learning model for feature representation. Graph Neural Network (GNN) is another model which learns the graph representations using specially designed neural layers and defines convolutions in the graph domain by aggregating feature information of each node from the connected neighbors. GraphSAGE [23] is proposed as the seminal spatial-based GNN framework that is founded upon the general notion of aggregator functions for the node embeddings. Zhang et al. [57] proposed a heterogeneous GNN model to consider heterogeneous graph structure and the heterogeneous contents together by collecting strongly correlated heterogeneous neighbors. Also, for better representation of heterogeneous graphs with rich node content features, in MAGNN method, Fu at al. [20], employs intra-meta-path aggregation for the content transformation of heterogeneous node attributes. Futhermore, Duran et. al. [17] proposed a method based Graph Convolutional Networks (GCN), which benefits from N-partite graphs to represent N-multiple user interaction contexts to learn node embeddings and addressed on recommender system for seamless integration of user contexts.

Our proposed location recommendation method focuses on heterogeneous LBSN graphs and offers a subgraph based node embedding technique. In this sense, it has similarity with the HIN embedding methods in the literature. Due to this, in the experiments, we provide comparison with a variety of such methods such as metapath2vec, HIN2Vec. GE and JLGE are also relevant to our work since they also challenge location recommendation problem and provide a network embedding based method. Therefore we included JLGE in our analysis for comparison. For the GNN based methods, most of the summarized techniques work on different graph structures than our study, mostly lacking either social network or location interaction. Among them, MAGNN is the most compatible one to our study, hence it is considered for comparison. In the literature, only few research efforts can be found that employ subgraphs in the embedding process for recommending locations to the users. Our approach differs from the previous works in the subgraph creation process. In the aforementioned studies, graphs are decomposed along with relationships or meta-paths. However, we decompose the graphs according to the geographical attributes of the nodes such that location nodes around the user's current positions with the specified radius and user nodes (friend, expert, trusted user) visiting these locations. With this approach, we aim to model users' preferences and interactions with respect to spatial context.

III. HETEROGENEOUS GRAPH EMBEDDING WITH SGWALK

In this section, we introduce our method SgWalk, a heterogeneous graph embedding method using random walk on user subgraphs rather than using the whole LBSN graph. Basically, we develop a graph embedding based solution combining random walks with a SkipGram-like model, which has been widely adopted in the literature and provides high accuracy performance on different tasks [16], [19], [22], [24], [38]. Our proposed method first constructs user subgraphs with respect to spatial contexts. This provides focusing on user preferences and relationships between users and locations for the spatial context and helps to eliminate unintended users and uninterested locations. Then random walk is applied over generated heterogeneous input graphs to obtain walk sequences to feed into SkipGram model for learning the node embedding. An illustration of these steps is given as an overview in Figure 1. In the rest of the section, we present the details of user subgraph construction strategy, node embedding learning process using SkipGram, and recommendation generation, after problem definition and preliminaries.

TABLE 1: Definitions Of Main Notations

Notation	Explanation
$\mathcal{G}$	Graph of location based social network
$\mathcal{U}$	Set of all users
$\mathcal{L}$	Set of all locations
$\mathcal{V}$	Set of all vertices as union of users and locations
$\mathcal{E}$	Set of edges representing connections between $v \in \mathcal{V}$
$\mathcal{W}$	Set of walk sequences
$\mathcal{A}$	List of all check-in activities
$\mathcal{A}_{u_i}$	List of check-in activities that belong to user u_i
$\mathcal{A}_{l_j}$	List of check-in activities that belong to location l_j
$\mathcal{R}_{u_i}$	Expert score of a user i
$\mathcal{R}_{l_j}$	Popularity score of a location j
r	Number of walks per node
s	Length of walk sequence
w	Window size for SkipGram process
d	Embedding vector dimension
k	Recommendation count from top

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access

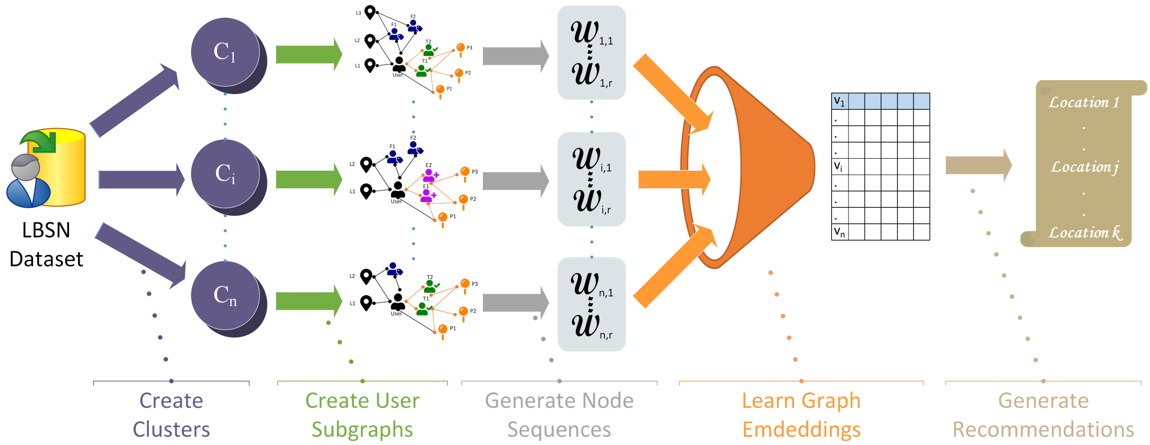


FIGURE 1: Overview of SgWalk

A. PRELIMINARIES

We can define personalized location recommendation problem as follows: Given a graph G corresponding to a particular LBSN, a user $u \in \mathcal{U}$ and the current location of the user, we aim to generate a set of locations $\{r_{l_1}, ..., r_{l_i}, ..., r_{l_n}\}$ that have not been visited by u as recommendations, such that each $r_{l_i} \in \mathcal{L}$ in the vicinity. The challenge is to populate this recommendation set with the highest accuracy. We follow four steps to fulfill this challenge: user subgraph construction, random walk generation, graph embedding, and location recommendation. Main notations used in this paper are given in Table 1. In the paper, capital letters are used for representing sets, small letters are used for constants value, vectors and matrix symbols are displayed with indices.

Definition 1 (Location)*:* A location is a specific site (e.g., a cafe or a movie theater) that can be uniquely identified. In our dataset, a location has two attributes: identifier (name), geographical location. We use l to represent a location and l^c to denote its corresponding geographical coordinates in terms of longitude and latitude.

Definition 2 (Check-in Activity)*:* A check-in activity is expressed as a quadruple $a(u,\ l,\ l^c,\ \tau)$ that denotes that in activity a, user u visits the location l on coordinates l^c at time τ.

Definition 3 (Vicinity)*:* Vicinity is a circular region defined by a user's current location p^u and the radius parameter ρ such that $| p^u - l^c | < \rho$. Obtained circular area is used as the recommendation region.

On the basis of the definitions for the basic LBSN concepts, we formally define the problem that we focus on in this study as follows.

Problem 1 (Location Recommendation):

Given a check-in activity set $\mathcal{A} = \{a_1, a_2, ..., a_n\}$, a querying user u_q in $\mathcal{U}$ within a vicinity, our goal is to find latent representations of v in $\mathcal{V}$ and generate a set of Top-N locations $\{l_1, ..., l_N\}$ as recommendations, so that the recommendation set consists of previously not visited locations u_q will be visited next time with the highest accuracy.

In this work, we devise a user subgraph based location recommender, and we hypothesize that utilizing user subgraphs of spatial context to generate random walk sequences improves the recommendation accuracy.

B. USER SUBGRAPH CONSTRUCTION

We employ an undirected and unweighted graph model to represent a particular subgraph. This graph, $\mathcal{G}$, is a tuple $\mathcal{G} < \mathcal{V}, \mathcal{E} >$ where $\mathcal{V}$ is a set of nodes v and $\mathcal{E}$ is a set of edges e. $\mathcal{V} \subset (\mathcal{U} \cup \mathcal{L})$ where $\mathcal{U}$ is a set of users and $\mathcal{L}$ is a set of locations. Note that $\mathcal{V}$ contains different subsets of $\mathcal{U}$ and $\mathcal{L}$. Moreover, it includes six different types of nodes: user, friend, expert, trusted user, location, and popular location. Node types in a subgraph is formally defined as follows:

Definition 4 (Friend)*:* Each user who takes part in an explicit direct relationship (friendship, following, etc.) in the social networks is considered as friend. Friendship relationship between two users u_i and u_j denoted as a tuple of (u_i, u_j).

Definition 5 (Trusted User)*:* A user is considered as trusted if the user's check-in activity behavior conforms to common behavior in visiting locations in a particular region.

Trustworthiness score calculation of a user is done according to the formulas stated in [8], which is adapted from the trust model developed in [35].

Definition 6 (Expert)*:* LBSN data typically do not include expert information explicitly. In this work, we employ a HITS-based [10] algorithm to determine the experts in the vicinity. In the HITS-based method, people who visit many high-quality locations in a region are regarded to have rich knowledge about the locations in the region. In the algorithm, users' expert scores are calculated by iteratively applying the formula in Equation 1. The users are sorted in decreasing order with respect to the score, and the top n% of the users are considered as experts.

Definition 7 (Popular Location)*:* As in expert information, LBSN datasets do not include any explicit information about

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access

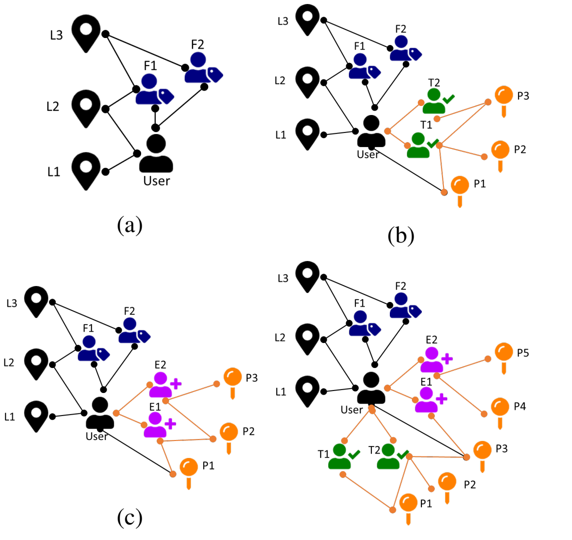
popularity of the locations. So we employ the same HITS-based [10] algorithm to find the popular locations in the vicinity, as well. In the algorithm, if a location is visited by many experts, it is more likely to be a popular location [6]. Location scores are calculated by iteratively applying the formula in Equation 2. The locations are sorted in decreasing order with respect to the score, and the top n% of the locations are considered as popular locations.

Expert user and popular location score calculation algorithms run simultaneously in an iterative manner as expressed in Equation 1 and Equation 2. In the equations, $\mathcal{R}_{u_i}$ denotes the expert score of user u_i, $\mathcal{R}_{l_j}$ denote the popularity score of the location l_j and $\mathcal{A}_{u_i}, \mathcal{A}_{l_j}$ denotes the check-in activities of user u_i or location l_j respectively.

$$\mathcal{R}_{u_i} = \left\| \sum_{l_j \in \mathcal{A}_{u_i}} \mathcal{R}_{l_j} \right\| \tag{1}$$

$$\mathcal{R}_{l_j} = \left\| \sum_{u_i \in \mathcal{A}_{l_j}} \mathcal{R}_{u_i} \right\| \tag{2}$$

In each iteration, popularity scores are updated by using the previous iteration's expert scores. In the same iteration, newly obtained popularity scores are employed for updating the new expert scores. At the end of each iteration, popularity scores and expert scores are normalized. Algorithm runs until the values converge.



(a)

(b)

(c)

(d)

FIGURE 2: Incremental Construction of User Subgraph

User subgraph is constructed incrementally according to the types of nodes. The simplest user subgraph that contains user, location and friend nodes is constructed by depending on her/his own and friends' check-in history (Figure 2a). It is further enriched by adding trusted users and popular locations visited by them (Figure 2b). In a similar way, experts and their visited popular locations are also included (Figure 2c). Finally, a complex user subgraph is constructed (Figure 2d) by using all introduced nodes, which are user, friend, location, expert, trusted user and popular locations. In order to analyze the effect of including such additional type of nodes, we conduct accuracy performance analysis experiments by using each subgraph structure separately.

In the user subgraph construction, four different contexts, such as personal, social, trustworthiness, and location (spatial) contexts, are considered to obtain improved recommendation results. The details of user subgraph construction are given in Algorithm 1. In this algorithm, userid and currentLoc represent the target user for recommendation and the user's current location for vicinity, respectively. In the algorithm, GetUserLocationsInVicinity procedure retrieves the previous check-ins of the current user in this region. Similarly, GetFriendLocationsInVicinity procedure gets the check-in history of the friends in this region. Likewise, GetExpertLocationsInVicinity and GetTULocationsInVicinity procedures find the check-ins of popular locations visited by experts and trusted users in the recommendation region, respectively. Following this, friends of the target user who have check-ins in the vicinity are retrieved in the GetFriendsOfUsermethod. In GetExpertsInVicinity and GetTrustedUsersInVicinity methods, Top-N experts and trusted users having check-ins in the vicinity are fetched. Once all those users are collected, the relationships between the current user and these users (i.e., friends, experts, and trusted users) are established. Finally, the visited location nodes and the edges between users and these locations are added to the subgraph. In the algorithm, trustedUserCount, expertCount, and popLocCount parameters limit the number of trusted users, experts, and popular locations in the subgraph.

C. RANDOM WALK WITH SGWALK

We perform random walks on the constructed user subgraphs, to sample from input heterogeneous information networks. Formally, for each user node u in $\mathcal{U}$, we initialize a random walk sequence starting from u until the walk length is reached. Similar to existing random walk based graph embedding techniques [16], [19], [24], [38], we generate a set of node sequences with length s by performing a specified number of random walks r on the subgraphs of each node u in $\mathcal{U}$. Algorithm 2 illustrates our random walk process to generate a set of walk sequences.

User subgraph constructed by Algorithm 1 is used within Algorithm 2. In this algorithm, walkLength and walkCount variables denote the maximum walk length and the number of sequences per subgraph, respectively. For each user, spatial context on which the latent representation to be constructed through random walks s is determined through FindClusters method. This method applies clustering algorithm DBSCAN [18] on the check-in activities $\mathcal{A}_u$ of user u with the radius parameter ρ. We prefer this algorithm since DBSCAN is based on neighborhood and density concepts and it does not require number of clusters in advance. While random walks are generated on the subgraphs, at each move, nextLocation

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access

Algorithm 1 User subgraph construction algorithm

```
procedure GenerateSubgraph($userid, currentLoc$)
Input: $userid$: id of user requesting recommendation
       $currentLoc$: geographical coordinate of the current position of user
Output: $G < V, E >$: User Subgraph
    trustedUserCount $\leftarrow$ the value denoting the top-n number of trusted users in the vicinity
    expertCount $\leftarrow$ the value denoting the top-n number of experts in the vicinity
    popLocCount $\leftarrow$ the value denoting the top-n number of popular locations in the vicinity
    $G < V, E > \leftarrow \emptyset$
    $vicinity \leftarrow$ GetUserLocationsInVicinity ($userid, currentLoc$)
    $vicinity \leftarrow vicinity \cup$ GetFriendLocationsInVicinity ($currentLoc$)
    $vicinity \leftarrow vicinity \cup$ GetExpertLocationsInVicinity ($currentLoc, popLocCount$)
    $vicinity \leftarrow vicinity \cup$ GetTULocationsInVicinity ($currentLoc, popLocCount$)
    $users \leftarrow$ GetFriendsOfUser ($userid$)
    $users \leftarrow users \cup$ GetExpertsInVicinity($currentLoc, expertCount$)
    $users \leftarrow users \cup$ GetTrustedUsersInVicinity($currentLoc, trustedUserCount$)
    for all $user$ in $users$ do
        $E \leftarrow E \cup$ new edge between $currentuser$ and $user$
    end for
    $V \leftarrow currentuser$
    $V \leftarrow V \cup users$
    for all $data$ in $vicinity$ do
        $V \leftarrow V \cup$ new location vertex for $data.location$
        $E \leftarrow E \cup$ new edge between $data.user$ and $data.location$
    end for
     return $G < V, E >$
end procedure
```

and nextUser variables store the location and the user nodes currently being visited by the random walk algorithm, respectively. In each iteration, nextLocation changes, and the visited node id is appended to the walk sequence. When the random walk length is reached, the algorithm returns the generated random walk sequence. Since our LBSN graph has unweighted and undirected edges from user to location, the transition probabilities are equal for all the nodes in every random walk step. Therefore, the next node is selected by uniformly sampling a node from the neighbors of the current node in both directions.

D. NODE EMBEDDING LEARNING WITH SKIPGRAM

Since the frequency distribution of vertices in random walks of social network and words in a language both follow a power-law [39], in our method, we use a technique similar to Word2Vec [34] to generate the low dimensional vector-space (d-dimensional) representation of a node in the graph. We adopt a Skip-Gram model with hierarchical softmax to generate the node embeddings. SkipGram is a model that maximizes the co-occurrence probability among the nodes that appear within a window with size w in the set of walk sequences, and hierarchical softmax is utilized to speed up the training phase. Formally, for a pair of nodes v_i and v_j appearing in the window in the set of walks $\mathcal{W}$, the co-occurrence probability is defined as given in Equation 3.

$$Pr((v_i, v_j) \in \mathcal{W}) = \Phi(\vec{v}_i \cdot \vec{v}_j) \tag{3}$$

In the equation $\Phi(\cdot)$, is the sigmoid function $\Phi(x) = 1/(1 + exp(-x))$, and $\vec{v}_i$ and $\vec{v}_j$ refer to the embeddings (vectors) of v_i and v_j , respectively. In addition, the Skip-Gram model employs negative edge sampling techniques to improve embedding accuracy by a randomly sampled edge between negative node $\tilde{v_k}$ and v_i that is not appearing in the set of walks. The negative edge probability is defined as given in Equation 4.

$$Pr((v_i, \tilde{v_k}) \notin \mathcal{W}) = 1 - Pr((v_i, \tilde{v_k}) \notin \mathcal{W}) = \Phi(-\vec{v}_i \cdot \vec{\tilde{v_k}}) \tag{4}$$

Here negative samples (nodes) are uniformly drawn from node distributions in the walks. In summary, for the pair of nodes (v_i, v_j), the SkipGram model maximizes the objective function as given in Equation 5.

$$O = log(\Phi(\vec{v}_i \cdot \vec{v}_j)) + \sum_{k=1,(v_i,\tilde{v_k})\notin\mathcal{W}}^{N} log(\Phi(-\vec{\tilde{v_k}} \cdot \vec{v}_i)) \tag{5}$$

In the equation, N is the number of negative edge samples. Parallel Asynchronous Stochastic Gradient Descent (ASGD) is utilized to learn the node embeddings efficiently by iterating over all node pairs appearing within a context window of size w in each walk sequence. The optimal value for

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access

Algorithm 2 Random walk generation algorithm

```
procedure GenerateRandomWalks()
    walkLength ← maximum number of node in a random walk sequence
    walkCount ← count of random walks sequences per user subgraph
    walks ← $\emptyset$
    for all user in $\mathcal{U}$ do
        userLocations ← FindClusters(user)
        for all curLocation in userLocations do
            $G < V, E >$← GenerateUserSubgraph(user, curLocation)
            nextLocation ← null
            while i < walkCount do
                walkSequence ← $\emptyset$
                while len(walkSequence) < walkLength do
                    nextUser ← SelectNextUser(nextLocation)
                    walkSequence.append(nextUser)
                    nextLocation ← SelectNextLocation(nextUser)
                    walkSequence.append(nextLocation)
                end while
                walks.append(walkSequence)
                i ← i + 1
            end while
        end for
    end for
     return walks                         ▷ returns the set of generated random walk sequences
end procedure
```

dimension size, d, of embedding vectors and the window size, w, (i.e., the context size) for model training using the Skip-gram model are determined through experiments whose details are given in the Experiments section (see Section IV-E).

E. LOCATION RECOMMENDATION

Once the graph embedding of the nodes in LBSN is performed, the vectorial representation of locations and users are learned in a shared space. For a given user $u \in \mathcal{U}$, we calculate the prediction score for each of the unvisited locations $l_j \in \mathcal{L}$ in the vicinity according to the Equation 6, and then we recommend top N of the ranked list of locations having the highest scores.

$$Prediction\_Score(u, l_j) = \vec{u}^T \cdot \vec{l_j} \tag{6}$$

In the equation $\vec{u}$ and $\vec{l_j}$ refer to the embeddings (vectors) of u and l_j, respectively.

IV. EXPERIMENTS

In this section, we present the experiments conducted for empirical evaluation of our approach, SgWalk. We first describe the datasets used in the experiments and evaluation metrics. Then evaluation strategy and parameter setting used in the analysis are presented. In the experiments, we firstly present analysis on the effect of including different node types in the subgraph for the recommendation accuracy. Then we give the evaluation results on the effects of two key parameters, vector dimension and window size, on the quality of the learnt embeddings. Then, we compare our proposed technique against the state-of-art methods. Statistical significance of the comparative results are presented as well. Lastly, validity threats and limitations of the proposed method are discussed.

A. DATASETS

For the evaluation, we use four real world datasets involving heterogeneous graphs, which are Brightkite [12], Foursquare [55], Gowalla [12] and Wee places [29] datasets. All of the datasets contain users' check-in data for the locations and the friends of users. In the experiments, we use the subset that is filtered for New York City. The main characteristics of used datasets are summarized in Table 2. As shown in the table, although these datasets have close average check-in counts per user and the average number of friends per, Foursquare has nearly double number of user, location, check-in and friendship counts compared to Brightkite.

TABLE 2: Dataset Characteristics

Dataset	Brightkite	Foursquare	Gowalla	Wee Places
No. of Users	6144	11154	10142	4812
No. of Locations	41870	103822	57756	28598
No. of Friendships	27138	59078	45618	31915
Check-ins per User	40.28	45.96	26.68	150.07
No. of Check-ins	247464	512645	270571	722119
Friends per User	4.42	5.30	4.50	6.63

In the datasets, some of the users tend to visit the same

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access

location more than once. This hinders the evaluation since recommending new locations to such users can never provide a hit count. To eliminate this problem, we simply grouped the records by user and location. In particular, for each user we use the latest check-in for a location if more than one check-in exists and ignore the others. The user data is sorted by check-in date from the oldest to recent, and partitioned into training and test datasets.

B. EVALUATION METRICS

We utilized three widely used metrics to evaluate the accuracy of SgWalk and baseline methods, which are precision@k, recall@k and f-measure@k [5], [14], [19], [33]. Precison@k is calculated by checking whether locations in the ground truth are ranked in the recommendation list of size k. It is calculated as given in Equation 7.

$$precision@k = \frac{\text{number of true locations in recommendation@k}}{\text{number of recommendations@k}} \quad (7)$$

Recall@k measures the ratio of truly recommended locations to the all visited locations in the ground truth, which is calculated as given in Equation 8.

$$recall@k = \frac{\text{number of true locations in recommendation@k}}{\text{number of locations in ground truth}} \quad (8)$$

Precison@k and Recall@k are inversely affected in evaluations. In other words, when Precison@k is high, Recall@k is low or vice-versa. To normalize these metrics f-measure is used, which is calculated as given in Equation 9.

$$f\text{-}Measure@k = \frac{2 \times Precision@k \times Recall@k}{Precision@k + Recall@k} \quad (9)$$

In the experiments, for all of the metrics, the accuracy is analyzed under k values of 3, 5, 8, 10, 15, 20.

C. EVALUATION METHODOLOGY AND PARAMETER SETTINGS

Since the challenged recommendation problem and the proposed solution involve spatial context, we need to acquire locations for the test users. To be able to simulate the current position of the user and locations in the vicinity, we cluster the check-in data with respect to locations. As the clustering algorithm, we use DBSCAN [18]. We prefer this algorithm since it does not require number of clusters in advance and it determines clusters based based on the concepts of neighborhood and density. To fulfill this, DBSCAN needs two parameters: the radius of neighborhood and the minimum number of points in a neighborhood. In the experiments, we set the minimum number of points as 3, and neighborhood radius as 2000m which is also consistent with the optimal neighborhood size in [32].

Once the clustering is performed, the center of each cluster is considered as the current location of the user to define the spatial context of the user and the vicinity for recommendation. The clustered user data is sorted by check-in date from old to recent. The sorted check-in data for each cluster are partitioned into training and test datasets such that the old data constitute the training data set, since we aim to find next location recommendations. Data is partitioned such that 80% of the data per user is used for training, 20% is for testing. The experiments are conducted under 5-fold cross validation. We set the walk count r to 10 per subgraph and the length of walk sequence s to 40 in random walk generation processes.

D. ANALYSIS ON THE EFFECT OF ADDING DIFFERENT NODE TYPES

As the first analysis, we compare the recommendation accuracy under the subgraphs A, B, C and, D illustrated in Figure 2a, 2b, 2c, and 2d, respectively, to assess the impact of adding new type of node. The results of the experiments performed on Brightkite, Foursquare, Gowalla and Wee Places datasets are given in Table 3. The results clearly indicate that types of the nodes affects the accuracy of recommendation in terms of precision, recall and f-measure metrics. Friendship relation exists on all datasets so we set it as the baseline subgraph. Then we investigate the impacts of 2 latent node types Expert and Trusted Users. As expected, subgraph having only friend nodes and its check-ins (subgraph A) has the lowest recommendation accuracy in the experiments among all subgraphs. It can be concluded that the friends of a user cannot cover enough number of location recommendations for user's current context.

Experiment results for subgraph B and subgraph C indicate that all subgraph acquire higher accuracy values than subgraph A. In addition, subgraph B performs better that subgraph C on all datasets. It is reasonable since if a location is visited by an expert, then it has higher possibility to be visited by the other users. On the other hand, trusted users may recommend essential places so its contribution is not as much as expert. As the last option, we consider the subgraph containing all node types together (subgraph D) and then we come up with the best accuracy in all datasets. Experts and trusted users have different check-in behavior and including these latent nodes improves the accuracy from popularity and trust-worthiness aspects respectively. For example, f-measure value is improved by 58% @5 recommendation in Wee Places dataset. These results also indicate that combining disjoint types (friend, expert, trusted user) improve the accuracy of the recommended locations.

In all datasets, we observe a similar accuracy pattern. The amount of accuracy gap varies, but the increase in the accuracy with inclusion of different types of nodes is consistent for all datasets. It is an important observation revealing that the accuracy of SgWalk is not dataset-specific. Although these four datasets' characteristics differ in terms of Check-ins per User and Friends per User values, Subgraph D, which contains friend, trusted user and expert nodes together, attains the best accuracy in all of the test cases. Furthermore, experiment results are more distinctive for a small number of recommendations but all subgraphs yield similar results by the increase of recommendation count. Subgraph D has

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access

TABLE 3: Recommendation accuracy for different subgraph structures

		Subgraph A			Subgraph B			Subgraph C			Subgraph D		
	@k	precision	recall	f-measure	precision	recall	f-measure	precision	recall	f-measure	precision	recall	f-measure
Brightkite	3	30,11	19,42	23,61	30.68	19.79	24.06	30.89	19.92	24.22	31.15	20.09	24.42
	5	21.01	22.57	21.76	21.75	23.37	22.53	21.72	23.34	22.50	22.22	23.88	23.02
	8	14.57	25.05	18.42	15.25	26.22	19.29	14.96	25.72	18.92	15.54	26.73	19.65
	10	12.04	25.89	16.44	12.57	27.03	17.16	12.43	26.73	16.97	12.90	27.73	17.61
	15	8.32	26.83	12.70	8.68	28.01	13.25	8.71	28.07	13.29	9.02	29.07	13.77
	20	6.39	27.46	10.37	6.61	28.44	10.73	6.62	28.47	10.74	6.92	29.75	11.23
Four Square	3	6.53	3.18	4.28	9.24	4.49	6.05	10.51	5.11	6.88	12.24	5.96	8.01
	5	4.90	3.97	4.39	6.77	5.49	6.06	7.75	6.28	6.94	9.28	7.52	8.31
	8	3.52	4.56	3.97	4.81	6.25	5.43	5.61	7.27	6.33	6.68	8.66	7.54
	10	2.98	4.83	3.68	4.07	6.61	5.03	4.71	7.62	5.81	5.61	9.11	6.94
	15	2.12	5.16	3.01	2.90	7.05	4.11	3.31	8.04	4.68	3.94	9.61	5.59
	20	1.64	5.32	2.51	2.22	7.21	3.39	2.52	8.16	3.85	3.02	9.81	4.62
Gowalla	3	6.53	3.18	4.28	9.24	4.49	6.05	10.51	5.11	6.88	12.24	5.96	8.01
	5	4.90	3.97	4.39	6.77	5.49	6.06	7.75	6.28	6.94	9.28	7.52	8.31
	8	3.52	4.56	3.97	4.81	6.25	5.43	5.61	7.27	6.33	6.68	8.66	7.54
	10	2.98	4.83	3.68	4.07	6.61	5.03	4.71	7.62	5.81	5.61	9.11	6.94
	15	2.12	5.16	3.01	2.90	7.05	4.11	3.31	8.04	4.68	3.94	9.61	5.59
	20	1.64	5.32	2.51	2.22	7.21	3.39	2.52	8.16	3.85	3.02	9.81	4.62
Wee Places	3	10.24	6.34	7.83	11.68	7.24	8.94	11.84	7.34	9.06	12.08	7.48	9.24
	5	7.19	7.42	7.30	8.20	8.47	8.33	8.38	8.65	8.51	8.53	8.80	8.66
	8	4.95	8.18	6.17	5.60	9.24	6.97	5.79	9.56	7.21	5.86	9.69	7.30
	10	4.11	8.49	5.54	4.65	9.59	6.26	4.81	9.94	6.48	4.85	10.01	6.53
	15	2.88	8.93	4.35	3.24	10.03	4.90	3.33	10.32	5.04	3.38	10.47	5.11
	20	2.23	9.19	3.59	2.48	10.25	3.99	2.55	10.54	4.11	2.60	10.74	4.19

TABLE 4: Recommendation accuracy (in f-measure) under varying window size *w* and dimension size *d* values

	brightkite@5					foursquare@5					gowalla@5					weeplaces@5				
w/d	100	120	150	**200**	250	100	120	150	**200**	250	100	120	150	**200**	250	100	120	150	**200**	250
1	2.98	3.01	3.17	3.18	3.11	0.22	0.24	0.26	0.27	0.31	0.49	0.49	0.49	0.51	0.56	0.52	0.54	0.55	0.63	0.63
2	27.48	27.51	27.54	**27.61**	27.48	11.3	11.77	11.77	**12.19**	12.05	10.44	10.48	10.62	**10.68**	10.65	6.01	6.47	6.96	**7.53**	7.48
3	20.72	21.3	21.76	21.85	21.79	8.03	7.99	7.9	7.95	7.91	8.48	8.56	8.60	8.87	8.81	5.33	5.34	5.35	5.38	5.37
4	23.4	22.66	22.92	23.21	23.34	9.2	9.19	9.03	8.86	9.14	8.94	8.81	8.99	8.88	8.82	5.29	5.42	5.50	5.20	5.12
5	18.78	18.65	18.88	18.94	18.68	6.48	6.57	6.29	6.3	6.19	7.13	7.41	7.47	7.57	7.51	4.36	4.21	4.07	3.52	3.17
6	19.91	20.11	20.04	20.08	19.98	6.99	7.08	7.06	7.09	6.92	7.63	7.66	7.39	7.51	7.67	3.81	3.76	3.55	3.14	2.72
7	17.65	17.81	17.58	17.23	17.52	5.77	5.75	5.82	5.52	5.72	6.78	6.69	6.77	6.70	6.84	3.31	3.05	2.77	2.33	2.1
8	18.62	18.56	18.26	18.39	18.07	6.36	6.06	6.4	6.29	6.07	6.52	6.66	6.63	6.68	6.68	2.80	2.70	2.50	2.05	1.94

its best f-measure value with recommendation count as 3 for Brightkite and Wee Places and also 5 for Foursquare and Gowalla and performance of top 5 in Brightkite and Wee Places is closer to top 3 than top 8. So we use top 5 as the recommendation count in the following experiments.

E. PARAMETER TUNING EXPERIMENTS

In this section, we present the results of tuning the parameters of our embedding model described in Section III-D. In particular, we examine the impact of window size w and the embedding vector dimension d when we train the model using Skip-gram language model. The candidate values of these two variables are w = 1, 2, 3, 4, 5, 6, 7, 8 and d = 100, 120, 150, 200, 250, respectively.

Here we examine the different values of the window size w and the dimension size d, i.e. (w, d) pairs, that provide recommendation accuracy in terms of f-measure. Table 4 shows SgWalk's f-measure values for the top 5 predictions for different values of w and d.

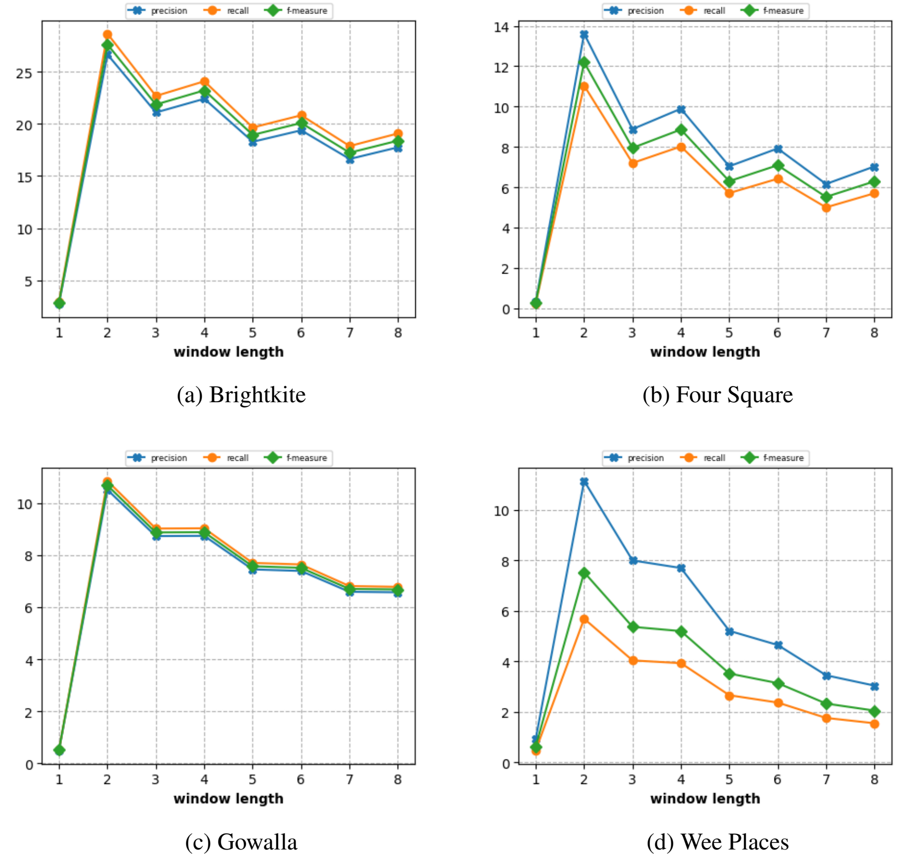
It is noticeable that as the dimension d increases until around 200, the accuracy increases as well. Increasing the dimension of vector has inverse affect after convergence point. Therefore, we set the dimension size 200 for all datasets.

On the other hand, we observe that the window size w has an interesting effect on the f-measure value. It converges alternatingly according to whether w is odd or even (Figure 3). This is due to the nature of the generated walk sequences which contains user and location nodes alternatingly. F-measure has the highest value at w=2 and drops and increases gradually as long as the window size w increases. Having lower w values also reduces the computational cost significantly. In Table 4, we highlight the parameter values, which give the best accuracy and are used for the rest of the evaluation.

In the parameter tuning experiments, the range of the the parameter values are determined manually. After the experiment runs are completed, we examine the accuracy results and check if the maximum value is obtained by the boundary values of the parameters. If so, we conduct new tests by increasing or decreasing the value of the parameter according to the boundary condition (being either min or max value in the previous experiment run). We continue until the parameter value that provides the maximum accuracy falls within the minimum and the maximum values (i.e. not a boundary value). For example, for vector dimension size tuning, we started with dimensions 100, 120, and 150, but

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access



(a) Brightkite

(b) Four Square

(c) Gowalla

(d) Wee Places

FIGURE 3: Recommendation accuracy (in f-measure) under varying window size (for dimension=200)

the maximum value was obtained at the boundary of the dimension size values. For this reason, we conducted new tests with dimension 200. Nevertheless, the maximum value was still obtained at the boundary. And then, we executed new tests with dimension 250. Finally, the dimension size setting providing the maximum accuracy fell inside the minimum and maximum boundaries. Therefore we concluded that the optimal dimension value is 200 for the SgWalk method.

F. COMPARATIVE RECOMMENDATION ACCURACY ANALYSIS

We analyze the accuracy performance of SgWalk under varying recommendation counts against the state-of-the-art graph embedding methods in the literature. HIN embedding is obtained in the literature in several ways, such as using homogeneous embedding techniques, employing meta-paths, bi-partite decomposition or using deep learning techniques. We select the baselines from each of these categories and also consider the techniques that use social relations. In the experimental setups, we use the default values for the parameters if it is not specified otherwise in the original work. The details of the compared methods are as follows:

- **DeepWalk [38]** learns node embeddings by first performing classical random walks on an input graph and then feeds the generated random walks to a SkipGram model. DeepWalk is originally designed for homogeneous graphs, so it is applied to a heterogeneous graph by ignoring the heterogeneity and treating all nodes and edges of the graph are in the same type. We set the number of walks per node r = 10, length of walk sequence s=40, and the window size the w = 10 and embedding vector dimension d = 100 for SkipGram.
- **PTE [44]** is a semi-supervised model for learning text embeddings using both labeled and unlabeled data. In these experiments, we used PTE in an unsupervised way. Specifically, for a heterogeneous graph, we created bipartite subgraphs as follows: (U)ser-(L)ocation, (U)ser-(F)riend, (U)ser-(T)rusted, (U)ser-(E)xpert, and then fed these graphs to PTE to generate the node embeddings.
- **metapath2vec [16]** generates meta-path guided random walks based on only a single predefined meta-path, and then feeds them to a SkipGram model. In the experiments with metapath2vec, we conducted analysis with four different meta-paths, which are "U-L-U" for representing different users visiting the same location, "U-F-L-U" for representing locations visited by friends of users, "U-T-L-U" for representing locations visited by trusted users, and "U-E-L-U" for representing locations visited by expert users. For random walk generation

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access

and the SkipGram process, we used the same values for the parameters r, s, w, and d as in the DeepWalk experiments.

- **HIN2Vec [19]** combines a set of meta-paths shorter than a certain length to perform meta-path guided random walks in order to learn both node embeddings and meta-path embeddings jointly. In these experiments, we set the maximum meta-path length to 3 and provided eight different edge types, which are "U-L", "L-U", "U-F", "F-L", "U-T", "T-L", "U-E" and "E-L". For random walk generation and the SkipGram process, we used the same values for the parameters r, s, w, and d as in the DeepWalk experiments.
- **JUST [24]** is a heterogeneous graph embedding technique using random walks with the jump and stay strategies to learn node embeddings more efficiently. In these experiments, we set stay or jump parameter across domains α as 0.5 and provide five domains as follows: (U)ser, (L)ocation, (F)riend, (T)rusted User, and (E)xpert User with the allowed transitions between domains. For random walk generation and the SkipGram process, we used the same values for the parameters r, s, w, and d as in the DeepWalk experiments.
- **JLGE [13]** is a spatial-temporal graph-based model that recommends location with learnt embeddings. It extends the LINE method on LBSN via multiple bi-partite graphs according to (U)ser, (L)ocation and (T)ime. In the experiments, we created six bi-partite graphs (U-U, U-L, U-T, L-U, L-L, L-T) as proposed in original work, for one month periods. We used the parameter values as reported in the original work such that d is set as 200, s is 100, ρ is 0.025, and negative edge count is 5.
- **MAGNN [20]** is Meta-path Aggregated Graph Neural Network for heterogeneous graph embedding. MAGNN first applies type-specific linear transformations for node content transformation via intra-meta-path aggregation, and utilizes inter-meta-path aggregation to generate node embeddings. In the experiments, we used eight different meta-paths, "U-L-U", "U-F-L-U", "U-T-L-U", "U-E-L-U" (same as in metapath2vec experiments) and the location counter parts of them as "L-U-L", "L-U-F-L", "L-U-T-L", "L-U-E-L". We set the best values for the GNN parameters (dropout rate, learning rate, the weight decay, etc.) as reported in the paper and used same value for the embedding dimension parameter d as in the previous experiments.

The results of the experiments performed on Brightkite, Foursquare, Gowalla and Wee Places datasets are given in Figure 4, Figure 5, Figure 6 and Figure 7, respectively. The results clearly indicate that SgWalk consistently performs better than the compared methods in terms of precision, recall, and f-measure metrics especially or recommendation count 8 and below. For the higher number of recommendations (10, 15 and 20), for majority of the results, we observe a higher accuracy results by SgWalk. As a summary, accuracy improvement with respect to the compared methods (in terms of f-measure under @5 recommendations) across the datasets are given in Table 5. It is observed that SgWalk provides the maximum improvement against DeepWalk with 102% on average. The minimum improvement is over JLGE with 23% on average. These results show a strong indication that SgWalk, focusing on spatial context and generating node embeddings by using subgraphs with respect to spatial context, can capture the contextual relationships more effectively. JLGE has the second-best accuracy value in the evaluation. However, JUST provides higher recommendation accuracy than JLGE and PTE on the Brightkite dataset and MAGNN has better results under high number of recommendation on Foursquare and Gowalla. JLGE, which is not a random walk based approach and designed for location recommendation, on average, performs better than other random walk based approaches JUST, metapath2vec, HIN2Vec, and DeepWalk. Among the meta-path based approaches MAGNN, metapath2vec and HIN2Vec, MAGNN is the best with the help of deep learning techniques and metapath2vec performs better than HIN2Vec under fewer number of recommendations. On the other hand, HIN2Vec first catches and then beats metapath2vec as the recommendation count increases. DeepWalk has the lowest accuracy since it is originally designed for homogeneous graphs.

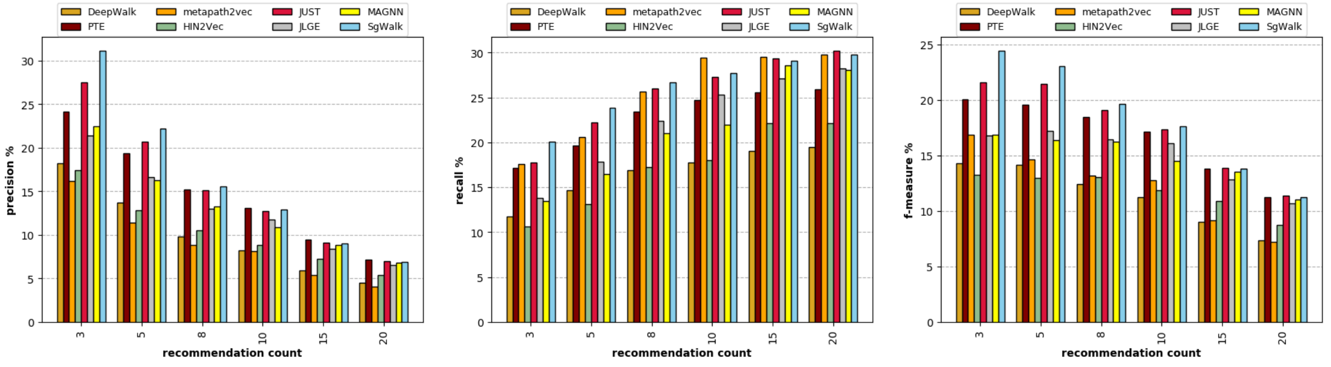
The accuracy of the techniques also varies depending on the characteristics of the datasets (given in Table 2). However, on all datasets, SgWalk attains the best accuracy values (for all f-measure values @10 or below, and majority of f-measure values @15 or above) Hence, we can conclude that the accuracy of SgWalk is not dataset dependent. When we analyze the accuracy of the state-of-the-art methods on datasets, we see that DeepWalk, HIN2VEC and MAGNN perform better on the Foursquare dataset, PTE and metapath2vec perform better on the Gowalla dataset, JUST and JLGE perform better on the Brightkite and Wee Places datasets, respectively. This shows the characteristics of the datasets has an effect on the methods. The results also show that *Check-ins per User* value affects the recommendation accuracy. The recommendation accuracy increases as the number of *Check-ins per User* increases. Moreover, when the number of recommendations is low, accuracy results are more distinctive, but with the increase in the number of recommendations, all techniques produce similar results.

TABLE 5: Improvement Percentages in f-Measure @5

	DeepWalk	PTE	Metapath2Vec	HIN2Vec	JUST	JLGE	MAGNN
Brightkite	63%	18%	57%	77%	7%	34%	40%
Four Square	105%	16%	39%	72%	95%	21%	34%
Gowalla	45%	21%	55%	28%	34%	23%	23%
Wee Places	197%	50%	77%	61%	102%	15%	71%
Average	**102%**	26%	57%	60%	60%	**23%**	42%

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access

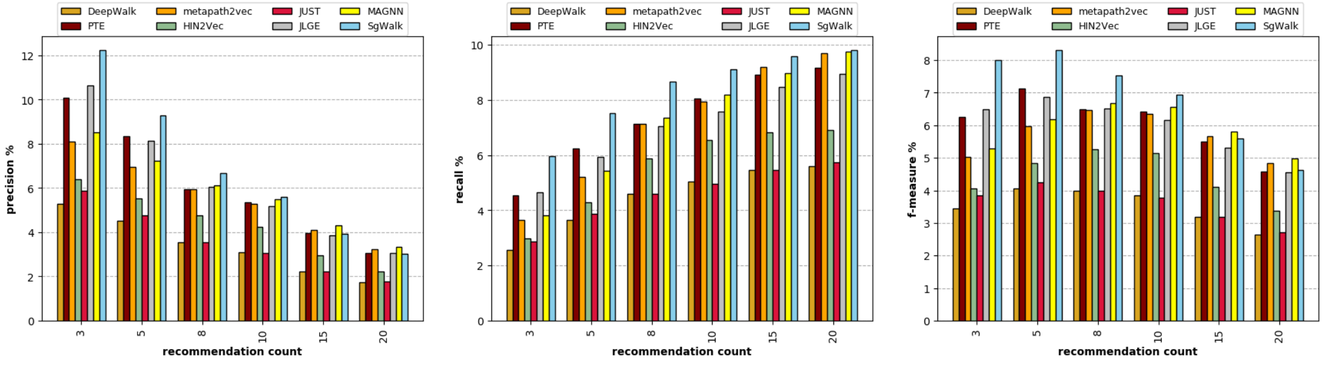


(a) Precision

(b) Recall

(c) f-Measure

FIGURE 4: Accuracy comparison of algorithms in terms of precision, recall and f-measure vs. number of recommendations for *Brightkite* dataset.

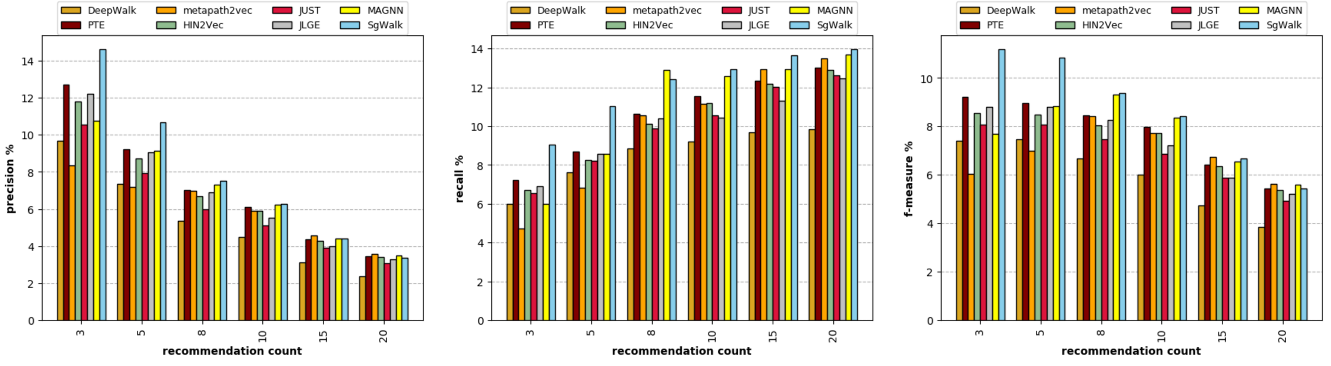


(a) Precision

(b) Recall

(c) f-Measure

FIGURE 5: Accuracy comparison of algorithms in terms of precision, recall, and f-measure vs. the number of recommendations for *Four Square* dataset.



(a) Precision

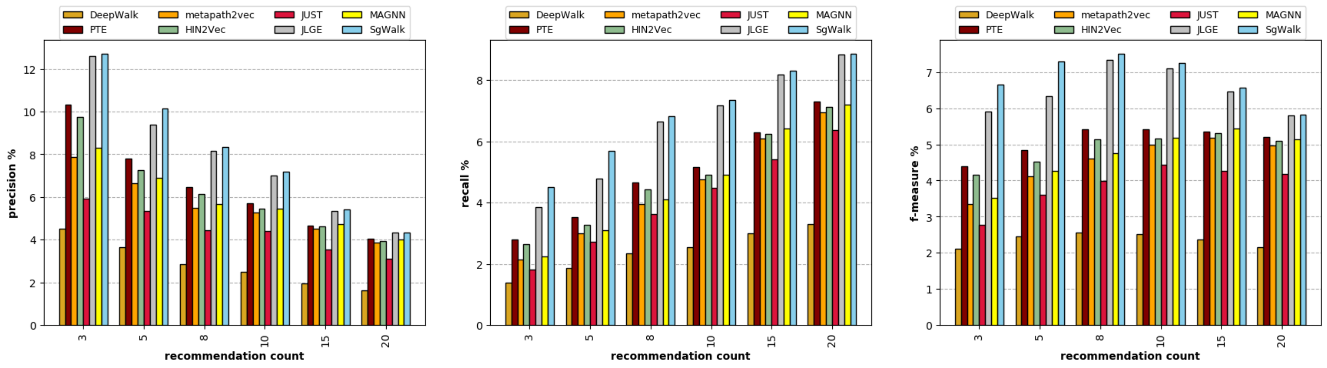
(b) Recall

(c) f-Measure

FIGURE 6: Accuracy comparison of algorithms in terms of precision, recall, and f-measure vs. the number of recommendations for *Gowalla* dataset.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access



(a) Precision

(b) Recall

(c) f-Measure

FIGURE 7: Accuracy comparison of algorithms in terms of precision, recall, and f-measure vs. the number of recommendations for *Wee Places* dataset.

G. STATISTICAL SIGNIFICANCE ANALYSIS FOR COMPARATIVE EVALUATION RESULTS

Statistical significance testing is a widely used analysis to assess the obtained performance improvements. In the literature, the t-test is the most commonly used method for statistical significance [42], [46], [47]. The t-test quantifies the difference between the performance values of the two samples. On the other hand, the p-value quantifies the probability of occurrence as or more extreme values assuming the null hypothesis, is true. A p-value larger than a chosen threshold indicates that the observation is not so unlikely to have occurred by chance. Therefore, we do not reject the null hypothesis of equal population means. If the p-value is smaller than the threshold, then we have evidence against the null hypothesis of equal population means. There are two types of t-tests, which are dependent and independent, according to the values that are obtained from the same or different environment. In this work, we evaluate all approaches with same dataset, so we use dependent t-test to assess the the statistical significance of our results.

Significance test is applied as follows: The difference of significance in results is investigated by using f-measure results. Our null hypothesis is that there is no difference in our proposed methods and the other compared approaches. A significance level is computed by considering f-measure value with respect to different recommendation counts in our experiments. Two-sided design is preferred to test whether the results are greater than or less than each other. Significance test is applied for each dataset separately. t-test score (t-value), and p-value are used to determine the significance[1]. When the significance level is low, the null hypothesis is rejected and it is proved the proposed approach achieves statistically significant improvements.

We used the significance level $\alpha = .05$ and find the critical value = 2.571 as the determination of the significance of SgWalk Results. t-test assessment results of SgWalk is given in Table 6 and significant values are marked with * in the table. SgWalk results are significantly different from DeepWalk and HIN2Vec approaches in all datasets. According to t-test scores, SgWalk results are statistically significant in 21 of 28 test cases covering 75% of the comparison results.

H. VALIDITY THREATS AND LIMITATIONS

SgWalk is a subgraph-based approach, and subgraphs are determined according to the check-ins clustered with respect to the spatial information. The size of the created clusters needs to be relatively small compared with the whole graph for SgWalk in order to capture the contextual relationships and to execute efficiently. Therefore, the selected clustering algorithm should generate the clusters from the dataset that allows small-sized user subgraph construction. We used the density-based algorithm DBSCAN to create clusters, and SgWalk has the best result with the four datasets used in experiments, with minimum 7% improvement over all cases under @5 recommendations. On the other hand, if the method generates large clusters failing to fulfill small-sized subgraphs construction, possibly due to the density distribution of the dataset, SgWalk may not capture contextual relationships and perform as expected.

Since clustering is a crucial step in SgWalk, it constitutes a vulnerable point whose quality can effect the outcome. In the proposed setting, since the challenged problem is location recommendation for LBSN, the clustering is built on spatial feature to extract the spatial context. For different settings and different contexts, other features of the social network can be user to decompose the full graph into small-sized subgraphs. For example, for a movie dataset, such subgraphs can be constructed by using genre (with sub-genres) feature, if a variety of genres are provided within the dataset.

V. CONCLUSION

In this work, we propose a location recommendation method, SgWalk, which makes use of user subgraph based node embeddings. In the hearth of the method lies a novel node embedding technique for HINs. The basic motivation in the

[1] We used a python script to calculate the t-value and p-value employing *ttest_rel* method from *scipy.stats* package.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access

TABLE 6: t-test Assessment Results of SgWalk

	Brightkite		Gowalla		Four Square		Wee Places	
	t-value	p-value	t-value	p-value	t-value	p-value	t-value	p-value
DeepWalk	7,041*	0,001*	7,992*	0.000*	7,769*	0,001*	23,021*	0,000*
PTE	2,074	0,093	2,807*	0,038*	2,647*	0,046*	6,150*	0,002*
JUST	1,728	0,145	8,724*	0.000*	4,19*	0,009*	8,270*	0,000*
metapath2Vec	8,334*	0,000*	2,100	0,090	1,912	0,114	5,674*	0,002*
HIN2Vec	4,443*	0,007*	5,242*	0,003*	2,817*	0,037*	6,071*	0,002*
JLGE	2,799*	0,038*	3,488*	0,017*	3,965*	0,011*	2,269	0,073
MAGNN	2,781*	0,039*	1,776	0,136	1,541	0,184	5,083*	0,004*

proposed embedding technique is to be able to capture interactions among the heterogeneous type of nodes according to spatial context. Node embeddings are popularly used in HIN related problems, and the embedding methods in the literature generally utilize bi-partite subgraphs and meta-paths for random walk generation, and use the full graph to generate walk sequences. Such approaches rely on meta-path structure assumptions and do not focus on contextual relationships. To be able to incorporate such relationships into the embedding, SgWalk makes use of contextual user subgraphs. It does not require any meta-path assumption, as well. The use of subgraphs, which are smaller structures including fewer nodes and edges than the full graph, is also helpful to improve computation time. The SgWalk technique follows four steps, which are (1) user subgraph construction (2) random walk sequences generation (3) learning graph embeddings (4) location recommendation generation, exploring the proximity between users and locations to provide recommendations.

In the proposed method, constructed user subgraph considers a wide range of information sources, including personal, social, spatial, and trustworthiness concepts. Such heterogeneous entities are frequently used in graph based recommendation methods. However the use of trust notion in location recommendation studies is scarce in the literature. Therefore, inclusion of trusted user nodes in the LBSN graph and using it the node embedding based setting is another novelty of the study.

We evaluate our location recommendation method in comparison to the state-of-the-art methods on well-known datasets filtered for New York City. The experiments show that our proposed algorithm, SgWalk, provides the highest accuracy against compared methods for majority of test cases, and the results indicate that our method brings minimum 23% improvement for location recommendation accuracy in terms of f-measure@5. The analysis on the results reveal that the obtained improvement by SgWalk is statistically significant (α = .05) for 75% of the test cases. The experimental results show a strong indication that SgWalk, focusing on spatial context and generating node embeddings by using subgraphs with respect to spatial context, can capture the contextual relationships more effectively, and hence it provides more accurate personalized and contextual location recommendations.

As future work, we plan to apply user-based subgraph generation in different domains such as movie and authorship databases for different recommendation tasks (such as movie recommendation to users and topic recommendation to authors). Additionally, studying the effect of temporal context nodes in the user subgraphs within the scope of LBSNs is another interesting research direction.

REFERENCES

[1] 41+ must know foursquare statistics in 2020. https://99firms.com/blog/foursquare-statistics/, 2021.

[2] Number of social media users in 2021/2022: Demographics & predictions. https://financesonline.com/number-of-social-media-users/, 2021.

[3] S Ahmadian, M Meghdadi, and M Afsharchi. Incorporating reliable virtual ratings into social recommendation systems. Applied Intelligence, 48(11):4448–4469, Nov 2018.

[4] I Ali, J Hong, and S Kim. Exploiting implicit and explicit signed trust relationships for effective recommendations. Proceedings of the Symposium on Applied Computing - SAC 17, 2017.

[5] H Bagci and P Karagoz. Context-aware location recommendation by using a random walk-based approach. Knowledge and Information Systems, 2015.

[6] J Bao, Y Zheng, and MF Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In Proceedings of the 20th International Conference on Advances in Geographic Information Systems, SIGSPATIAL '12, pages 199–208, New York, NY, USA, 2012. ACM.

[7] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. IEEE Transactions on Knowledge and Data Engineering, 30(9):1616–1637, 2018.

[8] D Canturk and P Karagoz. Trust-aware location recommendation in location-based social networks: A graph-based approach. Not published, 2020.

[9] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15, page 891–900, New York, NY, USA, 2015. Association for Computing Machinery.

[10] S Chakrabarti, B Dom, P Raghavan, S Rajagopalan, D Gibson, and J Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. Computer Networks and ISDN Systems, 30(1):65 – 74, 1998. Proceedings of the Seventh International World Wide Web Conference.

[11] Chen Cheng, Haiqin Yang, Irwin King, and Michael R. Lyu. A unified point-of-interest recommendation framework in location-based social networks. ACM Trans. Intell. Syst. Technol., 8(1), September 2016.

[12] E Cho, S Myers, and J Leskovec. Friendship and mobility: user movement in location-based social networks. in: Proceedings of the 17th acm sigkdd. international conference on Knowledge discovery and data mining. ACM, San Diego, pp 1082–1090, 2011.

[13] G. Christoforidis, P. Kefalas, A. Papadopoulos, and Y. Manolopoulos. Recommendation of points-of-interest using graph embeddings. In 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA), pages 31–40, 2018.

[14] P. Cui, X. Wang, J. Pei, and W. Zhu. A survey on network embedding. IEEE Transactions on Knowledge and Data Engineering, 31(5):833–852, 2019.

[15] Yves-Alexandre de Montjoye, César A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. Unique in the crowd: The privacy bounds of human mobility. Scientific Reports, 3(1):1376, Mar 2013.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access

[16] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. Metapath2vec: Scalable representation learning for heterogeneous networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17, page 135–144, New York, NY, USA, 2017. Association for Computing Machinery.

[17] Paula G. Duran, Alexandros Karatzoglou, Jordi Vitrià, Xin Xin, and Ioannis Arapakis. Graph convolutional embeddings for recommender systems. IEEE Access, 9:100173–100184, 2021.

[18] M Ester, H-P Kriegel, J Sander, and X Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, pages 226–231. AAAI Press, 1996.

[19] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM '17, page 1797–1806, New York, NY, USA, 2017. Association for Computing Machinery.

[20] Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding. In Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen, editors, WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020, pages 2331–2341. ACM / IW3C2, 2020.

[21] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Modeling temporal effects of human mobile behavior on location-based social networks. In Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, CIKM '13, page 1673–1678, New York, NY, USA, 2013. Association for Computing Machinery.

[22] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, page 855–864, New York, NY, USA, 2016. Association for Computing Machinery.

[23] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, page 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc.

[24] Rana Hussein, Dingqi Yang, and Philippe Cudré-Mauroux. Are meta-paths necessary? revisiting heterogeneous graph embeddings. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM '18, page 437–446, New York, NY, USA, 2018. Association for Computing Machinery.

[25] Y Jiang, H Chen, and B Yang. Deep social collaborative filtering by trust. In Proceedings of 2018 International Conference on Big Data Technologies, ICBDT '18, page 52–56, New York, NY, USA, 2018. Association for Computing Machinery.

[26] Bentian Li, Dechang Pi, Yunxia Lin, Izhar Ahmed Khan, and Lin Cui. Multi-source information fusion based heterogeneous network embedding. Information Sciences, 534:53 – 71, 2020.

[27] D Lian, C Zhao, X Xie, G Sun, E Chen, and Y Rui. Geomf: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, page 831–840, New York, NY, USA, 2014. Association for Computing Machinery.

[28] Nicholas Lim, Bryan Hooi, See-Kiong Ng, Xueou Wang, Yong Liang Goh, Renrong Weng, and Rui Tan. Origin-aware next destination recommendation with personalized preference attention. In WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021, pages 382–390. ACM, 2021.

[29] Y Liu, W Wei, A Sun, and C Miao. Exploiting geographical neighborhood characteristics for location recommendation. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM '14, pages 739–748, New York, NY, USA, 2014. ACM.

[30] Yuanfu Lu, Chuan Shi, Linmei Hu, and Zhiyuan Liu. Relation structure-aware heterogeneous information network embedding. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 4456–4463, 07 2019.

[31] Tianshu Lyu, Fei Sun, Peng Jiang, Wenwu Ou, and Yan Zhang. Compositional network embedding for link prediction. In Proceedings of the 13th ACM Conference on Recommender Systems, RecSys '19, page 388–392, New York, NY, USA, 2019. Association for Computing Machinery.

[32] M Mehaffy, S Porta, Y Rofè, and N Salingaros. Urban nuclei and the geometry of streets: The 'emergent neighborhoods' model. URBAN DESIGN International, 15(1):22–46, Mar 2010.

[33] S. N. Mighan, M. Kahani, and F. Pourgholamali. Poi recommendation based on heterogeneous graph embedding. In 2019 9th International Conference on Computer and Knowledge Engineering (ICCKE), pages 188–193, 2019.

[34] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, page 3111–3119, Red Hook, NY, USA, 2013. Curran Associates Inc.

[35] H-K Oh and S-W Kim. Identifying and exploiting trustable users with robust features in online rating systems. TIIS, 11(4):2171–2195, 2017.

[36] Osman Nuri Osmanli and İsmail Hakkı Toroslu. Using tag similarity in svd-based recommendation systems. In 2011 5th International Conference on Application of Information and Communication Technologies (AICT), pages 1–4. IEEE, 2011.

[37] Y Pan, F He, and H Yu. A correlative denoising autoencoder to model social influence for top-n recommender system. Frontiers of Computer Science, 14(3):143301, Dec 2019.

[38] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, page 701–710, New York, NY, USA, 2014. Association for Computing Machinery.

[39] Casper Petersen, Jakob Grue Simonsen, and Christina Lioma. Power law distributions in information retrieval. ACM Trans. Inf. Syst., 34(2), February 2016.

[40] Huseyin Polat and Wenliang Du. Svd-based collaborative filtering with privacy. In Proceedings of the 2005 ACM symposium on Applied computing, pages 791–795, 2005.

[41] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu. Heterogeneous information network embedding for recommendation. IEEE Transactions on Knowledge and Data Engineering, 31(2):357–370, 2019.

[42] Mark D. Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. In Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, CIKM '07, page 623–632, New York, NY, USA, 2007. Association for Computing Machinery.

[43] Lixin Su, Ruqing Zhang, Jiafeng Guo, Yixing Fan, Jiangui Chen, Yanyan Lan, and Xueqi Cheng. Beyond relevance: Trustworthy answer selection via consensus verification. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining, WSDM '21, page 562–570, New York, NY, USA, 2021. Association for Computing Machinery.

[44] Jian Tang, Meng Qu, and Qiaozhu Mei. PTE: predictive text embedding through large-scale heterogeneous text networks. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015, pages 1165–1174. ACM, 2015.

[45] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web, WWW '15, page 1067–1077, Republic and Canton of Geneva, CHE, 2015. International World Wide Web Conferences Steering Committee.

[46] Julián Urbano, Harlley Lima, and Alan Hanjalic. Statistical significance testing in information retrieval: An empirical analysis of type i, type ii and type iii errors. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'19, page 505–514, New York, NY, USA, 2019. Association for Computing Machinery.

[47] Julián Urbano, Mónica Marrero, and Diego Martín. A comparison of the optimality of statistical significance tests for information retrieval evaluation. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '13, page 925–928, New York, NY, USA, 2013. Association for Computing Machinery.

[48] F Wang, J She, Y Ohyama, W Jiang, G Min, G Wang, and M Wu. Maximizing positive influence in competitive social networks: A trust-based solution. Information Sciences, 546:559 – 572, 2020.

[49] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. In Collaborative Deep Learning for Recommender Systems, KDD '15, page 1235–1244, New York, NY, USA, 2015. Association for Computing Machinery.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3116226, IEEE Access

[50] Hongwei Wang, Fuzheng Zhang, Min Hou, Xing Xie, Minyi Guo, and Qi Liu. Shine: Signed heterogeneous information network embedding for sentiment link prediction. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM '18, page 592–600, New York, NY, USA, 2018. Association for Computing Machinery.

[51] Weiqing Wang, Hongzhi Yin, Xingzhong Du, Quoc Viet Hung Nguyen, and Xiaofang Zhou. Tpm: A temporal personalized model for spatial item recommendation. ACM Trans. Intell. Syst. Technol., 9(6), November 2018.

[52] Heitor Werneck, Nícollas Silva, Matheus Carvalho Viana, Fernando Mourão, Adriano C. M. Pereira, and Leonardo Rocha. A survey on point-of-interest recommendation in location-based social networks. In Proceedings of the Brazilian Symposium on Multimedia and the Web, WebMedia '20, page 185–192, New York, NY, USA, 2020. Association for Computing Machinery.

[53] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. Learning graph-based poi embedding for location-based recommendation. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM '16, page 15–24, New York, NY, USA, 2016. Association for Computing Machinery.

[54] B Yang, Y Lei, J Liu, and W Li. Social collaborative filtering by trust. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(8):1633–1647, 2017.

[55] D Yang, B Qu, J Yang, and P Cudre-Mauroux. Revisiting user mobility and social relationships in lbsns: A hypergraph embedding approach. In The World Wide Web Conference, WWW '19, pages 2147–2157, New York, NY, USA, 2019. ACM.

[56] 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, 29(11):2537–2551, Nov 2017.

[57] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. Heterogeneous graph neural network. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '19, page 793–803, New York, NY, USA, 2019. Association for Computing Machinery.

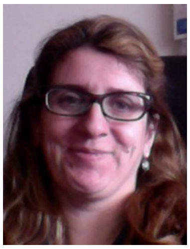
[58] S Zhang, L Yao, A Sun, and Y Tay. Deep learning based recommender system: A survey and new perspectives. ACM Comput. Surv., 52(1), February 2019.

[59] Weizhong Zhao, Huifang Ma, Zhixin Li, Xiang Ao, and Ning Li. Improving social and behavior recommendations via network embedding. Information Sciences, 516:125 – 141, 2020.

[60] V-W Zheng, Y Zheng, X Xie, and Q Yang. Collaborative location and activity recommendations with gps history data. In Proceedings of the 19th International Conference on World Wide Web, WWW '10, page 1029–1038, New York, NY, USA, 2010. Association for Computing Machinery.

[61] Y Zheng, X Xie, and W Ma. Geolife: a collaborative social networking service among user. location and trajectory. IEEE Data Eng Bull 33(2):32–40, 2010.

DENIZ CANTURK received the B.S. degree in Computer Engineering Department, Middle East Technical University (METU) Ankara, Turkey, in 2002. He received the M.S. degree in 2005 and the Ph.D.degree in 2021 from Computer Engineering Department, Middle East Technical University (METU) Ankara, Turkey. He has worked as a software engineer in different positions since 2002 in various companies. His current research interests include data mining, trust-aware recommendation systems and location-based social networks.



PINAR KARAGOZ is a full professor in Middle East Technical University (METU) Computer Engineering Department. She received her Ph.D. from the same department. During her doctoral studies, she worked as a researcher at SUNY Stony Brook University in New York, USA. She had research visits in MIT CSAIL, Ostrava University, Aalto University and Southern Denmark University. Her research focuses on data mining, machine learning algorithms, information retrieval, social media analysis and mining. She has publications in internationally recognized and indexed international journals including IEEE TKDE, ACM TWEB, IEEE TII and The Computer Journal, and she has about 100 papers in international conferences in the area. In 2016 she received the best paper award in IEEE Transactions on Industrial Informatics. In 2017, her paper was nominated for Wilkes Award of the Computer Journal.

• • •

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/